ARIB STD-T63-26.346 V11.7.0

Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs

(Release 11)

Refer to “Industrial Property Rights (IPR)” in the preface of ARIB STD-T63 for Related Industrial Property Rights. Refer to “Notice” in the preface of ARIB STD-T63 for Copyrights.

3GPP TS 26.346 V11.7.0 (2013-12)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Multimedia Broadcast/Multicast Service (MBMS);
Protocols and codecs
(Release 11)

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.

The present document has not been subject to any approval process by the 3GPP Organizational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organizational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

UMTS, IP, packet mode, codec, broadcast

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© 2013, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).
All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners
LTE™ is a Trade Mark of ETSI currently being registered for the benefit of its Members and of the 3GPP Organizational Partners
GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword 9
Introduction 9
1 Scope 10
2 References 10
3 Definitions and abbreviations 14
3.1 Definitions 14
3.2 Abbreviations 15
4 MBMS system description 16
4.1 MBMS functional layers 16
4.2 MBMS User Service Entities 17
4.3 MBMS bearer service architecture 18
4.4 Functional Entities to support MBMS User Services 18
4.4.1 MBMS User Service Architecture 18
4.4.1a Content Provider / Multicast Broadcast Source 21
4.4.2 MBMS Key Management Function 21
4.4.3 MBMS Session and Transmission Function 21
4.4.4 User Service Discovery / Announcement function 21
4.4.5 Interactive Announcement Function 22
4.4.6 MBMS UE 22
4.5 Usage of identity of MBMS session 22
4.6 Time Synchronization between the BM-SC and MBMS UEs 23
5 Procedures and protocols 24
5.1 Introduction 24
5.2 User Service Discovery/Announcement 24
5.2.1 Introduction 24
5.2.2 MBMS User Service Description Metadata Fragments 24
5.2.2.1 Introduction 24
5.2.2.2 Session Description 26
5.2.2.3 Associated Delivery Procedure Description 26
5.2.2.4 Service Protection Description 26
5.2.2.5 FEC Repair Stream Description 27
5.2.2.6 Media Presentation Description 27
5.2.2.7 Schedule Description 27
5.2.2.8 Filter Description 28
5.2.3 User Service Announcement over a MBMS bearer 28
5.2.3.1 General 28
5.2.3.2 Metadata Envelope Transport 28
5.2.3.3 Metadata Envelope and Metadata Fragment Association with FLUTE 28
5.2.4 User Service Announcement using Interactive Announcement Function 29
5.2.5 User Service Announcement over point-to-point push bearers. 29
5.2.5.1 General 29
5.2.5.2 Supported Metadata Syntaxes 29
5.2.5.3 Consistency Control and Syntax Independence 29
5.2.5.4 Metadata Envelope Definition 29
5.2.5.5 Delivery of the Metadata Envelope 29
5.2.5.6 Metadata Envelope Transport 30
5.2.5.7 User service announcement over SMS bearers 30
5.2.5.8 User service announcement over HTTP push bearers 30
5.2.6 Metadata Fragment Encapsulation to aggregate Service Announcement Documents 30
5.2.7 Registration and Deregistration Procedure for MBMS User Service Consumption 30
5.3 User Service Initiation/Termination 31
5.3.1 Initiation of MBMS Bearer Service based Services 31
5.3.2 Termination of MBMS Bearer Service based Services 32

5.3.3 Initiation of Unicast Bearer Service based Services 33
5.3.4 Termination of Unicast Bearer Service based Services....33
5.3.5 Scalable Service Initiation and Termination for MBMS Services 34
5.3.5.1 General 34
5.3.5.2 Randomization of Service Initiation over Time 34
5.3.5.3 Randomization of Service Termination over Time 34
5.4 MBMS Data Transfer Procedure 35
5.4.1 MBMS Data Transfer Procedure using MBMS Bearer Services....35
5.4.2 MBMS Data Transfer Procedure using other UMTS Bearer Services....36
5.5 MBMS Protocols 36
5.6 3GP-DASH and MBMS....36

6 Introduction on Delivery Methods....37

7 Download Delivery Method....38
7.1 Introduction 38
7.2 FLUTE usage for MBMS download 38
7.2.1 Fragmentation of Files 39
7.2.2 Symbol Encoding Algorithm....39
7.2.3 Blocking Algorithm 39
7.2.4 Congestion Control....39
7.2.5 Content Encoding of Files for Transport....40
7.2.6 Transport File Grouping....40
7.2.7 Signalling of Parameters with Basic ALC/FLUTE Headers 40
7.2.8 Signalling of Parameters with FLUTE Extension Headers 40
7.2.9 Signalling of Parameters with FDT Instances 41
7.2.10 FDT Schema....42
7.2.10.1 Extended FLUTE FDT Schema 42
7.2.10.2 3GPP FDT Extension Type Schema....43
7.2.10.3 IETF FDT Schema 44
7.2.10.4 Example of FDT....45
7.2.10.5 3GPP FDT Extensions....45
7.2.10.6 FEC Redundancy Level Extension....46
7.2.11 MBMS Session Identity....46
7.2.12 FEC Scheme definition 46
7.2.12.1 General 46
7.2.13 Caching Directives....46
7.2.14 Indicating a full FDT snapshot 47
7.2.15 Decryption key indicating of protected download data....47
7.3 SDP for Download Delivery Method....48
7.3.1 Introduction....48
7.3.2 SDP Parameters for MBMS download session 48
7.3.2.1 Sender IP address 48
7.3.2.2 Number of channels....49
7.3.2.3 Destination IP address and port number for channels....49
7.3.2.4 Transport Session Identifier (TSI) of the session 49
7.3.2.5 Multiple objects transport indication 49
7.3.2.6 Session Timing Parameters....49
7.3.2.7 Mode of MBMS bearer per media....50
7.3.2.8 FEC capabilities and related parameters....50
7.3.2.9 Service-language(s) per media 51
7.3.2.10 Bandwidth Specification....51
7.3.3 SDP Examples for FLUTE Session 51
7.4 OMA Push usage for MBMS Download....51
7.4.1 Introduction....51
7.4.2 HTTP registration and deregistration procedure....51
7.4.3 MBMS Download Delivery Method over OMA push bearers 52
7.5 FLUTE session setup and control with RTSP....53
7.5.1 Introduction....53
7.5.2 SDP handling 53
7.5.3 RTSP SETUP Method 53
7.5.4 RTSP PLAY Method....53

7.5.5 RTSP PAUSE Method 54
7.5.6 RTSP TEARDOWN Method 54

8 Streaming delivery method 54
8.1 Introduction 54
8.2 Transport protocol 54
8.2.1 RTP payload formats for media 54
8.2.2 FEC mechanism for RTP 55
8.2.2.0 General 55
8.2.2.1 Sending Terminal Operation (Informative) 56
8.2.2.2 Receiving Terminal Operation (Informative) 57
8.2.2.3 (Void) 57
8.2.2.4 Packet format for FEC source packets 57
8.2.2.5 Packet Format for Repair packets 58
8.2.2.6 Void 58
8.2.2.7 FEC block Construction algorithm and example (informative) 58
8.2.2.8 Void 59
8.2.2.9 Source FEC Payload ID 59
8.2.2.10 Repair FEC payload ID 59
8.2.2.10a FEC Object Transmission information 59
8.2.2.11 Hypothetical FEC Decoder 60
8.2.2.12 Void 60
8.2.2.13 Signalling 60
8.2.2.13a SDP for FEC source packet streams 61
8.2.2.14 SDP for FEC repair packet streams 61
8.2.2.15 Signalling example for FEC 61
8.2.3 General RTP Header Extension Mechanism 64
8.2.3.1 Introduction 64
8.2.3.2 Timestamp Offset 64
8.3 Session description 65
8.3.1 SDP Parameters for MBMS streaming session 65
8.3.1.1 Sender IP address 65
8.3.1.2 Destination IP address and port number for channels 65
8.3.1.3 Media Description 66
8.3.1.4 Session Timing Parameters 66
8.3.1.5 Mode of MBMS bearer per media 66
8.3.1.6 Service-language(s) per media 66
8.3.1.7 Bandwidth specification 66
8.3.1.8 FEC Parameters 66
8.3.1.9 FEC Flow ID attribute 67
8.3.1.10 Buffer Requirement Signaling 68
8.3.1.11 Interleaving Signaling 68
8.3.2 SDP Example for Streaming Session 68
8.3.2.1 SDP Description for QoE Metrics 69
8.3.2.2 OMA-DM Configuration of QoE Metrics 69
8.4 Quality of Experience 74
8.4.1 General 74
8.4.2 QoE Metrics 74
8.4.2.1 Corruption duration metric 75
8.4.2.2 Rebuffering duration metric 76
8.4.2.3 Initial buffering duration metric 76
8.4.2.4 Successive loss of RTP packets 76
8.4.2.5 Frame rate deviation 76
8.4.2.6 Jitter duration 77
8.4.2.7 Content Access/Switch Time 77
8.4.2.8 Network Resource 77
8.4.2.9 Average codec bitrate 78
8.4.2.10 Codec information 78
8.4.2.11 Loss of Objects 78
8.4.2.12 Distribution of Symbol Count Underrun for Failed Blocks 79
8.4.3 Example metrics initiation with SDP 79
8.5 Using MBMS Streaming delivery on Unicast 80

9 Associated delivery procedures 80
9.1 Introduction 80
9.2 Associated Procedure Description 81
9.3 File Repair Procedure 82
9.3.1 Introduction 82
9.3.2 Starting Time of the Associated Delivery Procedure for MBMS Download Delivery 82
9.3.3 Identification of Missing Data from an MBMS Download 83
9.3.4 Back-off Timing the Procedure Initiation Messaging for Scalability 84
9.3.4.1 Offset time 84
9.3.4.2 Random Time Period 84
9.3.4.3 Back-off Time 84
9.3.4.4 Reset of the Back-off Timer 84
9.3.5 File Repair Server Selection 85
9.3.5.1 List of Server URIs 85
9.3.5.2 Selection from the Server URI List 85
9.3.6 File Repair Request Messages 85
9.3.6.0 General 85
9.3.6.1 Symbol-Based File Repair Request Message Format 86
9.3.6.2 Byte-Range-Based File Repair Request Message Format 88
9.3.7 File Repair Response Message 89
9.3.7.1 Symbol-Based File Repair Response Messages Codes 89
9.3.7.2 Symbol-Based File Repair Response Message Format for HTTP Carriage of Repair Data 91
9.3.7.2a Byte-Range-Based File Repair Response Message Format for HTTP Carriage of Repair Data 92
9.3.7.3 File Repair Response for Broadcast/Multicast of Repair Data 92
9.3.7.4 File Repair Response Message Format for HTTP carriage of Complete Files 93
9.3.8 Server Not Responding Error Case 93
9.4 The Reception Reporting Procedure 94
9.4.1 Identifying Complete File Reception from MBMS Download and Determining Download Status 94
9.4.2 Identifying Complete MBMS Delivery Session Reception 94
9.4.3 Determining Whether a Reception Report Is Required 95
9.4.4 Request Time Selection 95
9.4.5 Reception Report Server Selection 96
9.4.6 Reception Report Message 96
9.4.7 Reception Report Response Message 98
9.5 XML-Schema for Associated Delivery Procedures 98
9.5.1 Generic Associated Delivery Procedure Description 98
9.5.2 Example Associated Delivery Procedure Description Instance 98
9.5.3 XML Syntax for a Reception Report Request 99
9.5.3.1 Use of Specific Values 100
9.5.3.2 Example XML for the Reception Report Request 100

10 Media codecs and formats 101
10.1 General 101
10.2 Speech 101
10.3 Audio 101
10.4 Synthetic audio 102
10.5 Video 102
10.6 Still images 103
10.7 Bitmap graphics 104
10.8 Vector graphics 104
10.9 Text 104
10.10 Timed text 104
10.11 3GPP file format 105
10.12 Scene Description 105
10.13 Timed graphics 105

11 MBMS Metadata 105
11.1 The MBMS Metadata Envelope 105
11.1.1 Supported Metadata Syntaxes 105
11.1.2 Consistency Control and Syntax Independence 105
11.1.3 Metadata Envelope Definition 106
11.1.4 Delivery of the Metadata Envelope 107

11.2 MBMS User Service Description Metadata Fragment 107
11.2.1 Definition of the MBMS User Service Bundle Description 107
11.2.1.1 Initial Definition 107
11.2.1.2 Extensions to the User Service Bundle Description 109
11.2.2 Example MBMS User Service Description Instances 112
11.2A Schedule Description Metadata Fragment 115
11.2A.1 Procedures for Schedule Description Metadata Fragment 115
11.2A.2 XML-Schema for the Schedule Description Meta Data Fragment 116
11.2A.3 Examples of the Schedule Description Meta Data Fragment 118
11.2B Filter Description Metadata Fragment 122
11.2B.1 Introduction 122
11.2B.2 Procedures for Filter Description Metadata Fragment 122
11.2B.3 Usage of Location Filter 122
11.2B.4 XML Schema for the Filter Description Metadata Fragment 123
11.3 Service Protection Description Metadata Fragment 125
11.3.1 Definition of the Protection Description 125
11.3.2 Example of a Security Description: 127
11.4 Service Protection Registration Format 127
11.4.1 Data Format 127
11.4.2 Example 127
11.5 Service Protection De-Registration Format 128
11.5.1 Data Format 128
11.5.2 Example 128
11.6 Service Protection MSK Request Format 128
11.6.1 Data Format 128
11.6.2 Example 129
11.7 Service Protection Registration and De-Registration Response Format 129
11.7.1 Data Format 129
11.7.2 Example 129
11.8 Service Protection MSK Response Format 130
11.8.1 Data Format 130
11.8.2 Example 130
11.9 MBMS Feature Requirements 131

Annex A (normative): FLUTE Support Requirements 132

Annex B (normative): FEC encoder specification 134

B.1 Void 134

B.2 Void 134

B.3 File download 134
B.3.1 Void 134
B.3.2 Void 134
B.3.3 Void 134
B.3.4 Example parameters 134
B.3.4.1 Parameter derivation algorithm 134
B.3.4.2 Examples 134

B.4 Streaming 135
B.4.1 Void 135
B.4.2 Void 135
B.4.3 Void 135
B.4.4 Example parameters 135
B.4.4.1 Parameter derivation algorithm 135
B.4.4.2 Examples 136

B.5 Void 136

B.6 Void 136

B.7 Void 136

B.8 Void 136

Annex C (informative): IANA registration 137
C.1 Registration of SDP Protocol Identifiers for Source packet 137
C.2 Registration of SDP Protocol identifier for repair packets 137
C.3 Registration of MIME type "application/simpleSymbolContainer" 137
C.4 Registration of MIME type "application/mbms-user-service-description+xml " 138
C.5 Registration of MIME type "application/mbms-envelope+xml " 139
C.6 Registration of MIME type "application/mbms-protection-description+xml " 141
C.7 Registration of MIME type "application/mbms-associated-procedure-description+xml" 142
C.8 Registration of MIME type "application/mbms-msk+xml" 143
C.9 Registration of MIME type "application/mbms-register+xml" 144
C.10 Registration of MIME type "application/mbms-deregister+xml" 145
C.11 Registration of MIME type "application/mbms-reception-report+xml" 146
C.12 Registration of MIME type "application/mbms-msk-response+xml" 147
C.13 Registration of MIME type "application/mbms-register-response+xml" 148
C.14 Registration of MIME type "application/mbms-schedule+xml" 149
C.15 Registration of MIME type "application/mbms-filter+xml" 150

Annex D (informative): RTP packetization guidelines 152

Annex E (informative): Void 153

Annex F (informative): Hybrid Streaming Delivery 154
F.1 Introduction 154
F.2 Switching between MBMS and PSS Access 154
F.2.1 Synchronization of Flows 154
F.2.2 Switching from MBMS to PSS Access 154
F.2.3 Switching from PSS to MBMS Access 155

Annex G (informative): Guidelines for Channel Tune-in and Switch Time Reduction 156
G.1 Interleaving for Tune-in Time Reduction of FEC protected MBMS Services 156
G.1.1 Timestamp Offsets 156
G.1.2 Early Playout 157
G.2 FEC Stream & Channel Bundling 157
G.2.1 Introduction 157
G.2.1.1 Full Channel Bundling (All flows of several user services) 157
G.2.1.2 No Bundling 158
G.2.1.3 Partial Channel Bundling (Some flows of several user services) 158
G.2.1.3 Stream Bundling (All flows of a single user services) 158

Annex H (informative): QoE Reporting Management Object Device Description Framework ... 159

Annex I (informative): Void 164

Annex J (normative): Schema 164
J.1 User Service Description schema 164
J.2 Version and Delimiter schema 166

Annex K (informative): Change history 167

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

MBMS is a point-to-multipoint service in which data is transmitted from a single source entity to multiple recipients. Transmitting the same data to multiple recipients allows network resources to be shared.

The MBMS bearer service offers two modes:

- Broadcast Mode.
- Multicast Mode.

MBMS user services can be built on top of the MBMS bearer service. Further, the MBMS User Services may use other UMTS bearer services to deliver data. The present document specifies two delivery methods for the MBMS user services: download and streaming. Examples of applications using the download delivery method are news and software upgrades. Delivery of live music is an example of an application using the streaming delivery method.

There can be several MBMS user services. The objective of the present document is the definition of a set of media codecs, formats and transport/application protocols to enable the deployment of MBMS user services. The present document takes into consideration the need to maximize the reuse of components of already specified services like PSS and MMS.

1 Scope

The present document defines a set of media codecs, formats and transport/application protocols to enable the deployment of MBMS user services either over the MBMS bearer service or other UMTS Bearer Services within the 3GPP system.

In this version of the specification, only MBMS download and streaming delivery methods are specified. The present document does not preclude the use of other delivery methods.

The present document includes information applicable to network operators, service providers and manufacturers.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 22.146: "Multimedia Broadcast/Multicast Service; Stage 1".

[3] 3GPP TS 22.246: "Multimedia Broadcast/Multicast Service (MBMS) user services; Stage 1".

[4] 3GPP TS 23.246: "Multimedia Broadcast/Multicast Service (MBMS); Architecture and functional description".

[5] 3GPP TS 25.346: "Introduction of Multimedia Broadcast/Multicast Service (MBMS) in the Radio Access Network (RAN); Stage 2".

[6] IETF STD 0064/RFC 3550 (July 2003): "RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson.

[7] IETF STD 0006/RFC 0768 (August 1980): "User Datagram Protocol", J. Postel.

[8] IETF STD 0005/RFC 0791 (September 1981): "Internet Protocol", J. Postel.

[9] IETF RFC 3926 (October 2004): "FLUTE - File Delivery over Unidirectional Transport", T. Paila, M. Luby, R. Lehtonen, V. Roca, R. Walsh.

[10] IETF RFC 3450 (December 2002): "Asynchronous Layered Coding (ALC) Protocol Instantiation", M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, J. Crowcroft.

[11] IETF RFC 3451 (December 2002): "Layered Coding Transport (LCT) Building Block", M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, M. Handley, J. Crowcroft.

[12] IETF RFC 5052 (August 2007): "Forward Error Correction (FEC) Building Block", M. Luby, M. Watson, L. Vicisano.

[13] IETF RFC 3695 (February 2004): "Compact Forward Error Correction (FEC) Schemes", M. Luby, L. Vicisano.

[14] IETF RFC 4566 (July 2006): "SDP: Session Description Protocol", M. Handley, V. Jacobson and C. Perkins.

[15] IETF RFC 4570 (July 2006): "Session Description Protocol (SDP) Source Filters", B. Quinn, R. Finlayson.

[16] Void.

[17] IETF RFC 3048 (January 2001): "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer", B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, M. Luby.

[18] IETF RFC 2616 (June 1999): "Hypertext Transfer Protocol -- HTTP/1.1".

[19] IETF STD 0066/RFC 3986 (January 2005): "Uniform Resource Identifier (URI)".

[20] 3GPP TS 33.246: "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)".

[21] OMG: "Unified Modeling Language (UML), version 1.5" (formal/03-03-01).

[22] W3C Recommendation 28 October 2004: "XML Schema Part 2: Datatypes Second Edition".

[23] IETF RFC 5234 (January 2008): "Augmented BNF for Syntax Specifications: ABNF", D. Crocker and P. Overell.

[24] 3GPP TS 26.290: "Audio codec processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec; Transcoding functions".

[25] 3GPP TS 26.304: "Floating-point ANSI-C code for the Extended Adaptive Multi-Rate - Wideband (AMR-WB+) codec".

[26] 3GPP TS 26.273: "Speech codec speech processing functions; Extended Adaptive Multi-Rate - Wideband (AMR-WB+) speech codec; Fixed-point ANSI-C code".

[27] Void.

[28] 3GPP TS 26.401: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; General description".

[29] 3GPP TS 26.410: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Floating-point ANSI-C code".

[30] 3GPP TS 26.411: "General audio codec audio processing functions; Enhanced aacPlus general audio codec; Fixed-point ANSI-C code".

[31] W3C Recommendation 04 February 2004: "Extensible Markup Language (XML) 1.1", T. Bray, J. Paoli, C. Sperberg-McQueen, E. Maler, F. Yergeau and J. Cowan.

[32] 3GPP TS 26.244: "Transparent end-to-end streaming service; 3GPP file format (3GP)".

[33] IETF RFC 4867 (April 2007): "RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs", J. Sjoberg, M. Westerlund, A. Lakaniemi, Q. Xie.

[34] IETF RFC 4352 (January 2006): "RTP Payload Format for the Extended Adaptive Multi-Rate Wideband (AMR-WB+) Audio Codec", Sjoberg J. et al.

[35] IETF RFC 6184 (2011): "RTP Payload Format for H.264 Video", Y.-K. Wang, R. Even, T. Kristensen, R. Jesup.

[36] Void.

[37] IETF RFC 2557 (March 1999): "MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)", J. Palme, A. Hopmann, N. Shelness.

[38] IETF RFC 3890 (September 2004): "A Transport Independent Bandwidth Modifier for the Session Description Protocol (SDP)", M. Westerlund.

[39] IETF RFC 3556 (July 2003): "Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth", S. Casner.

[40] 3GPP TS 24.008: "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3".

[41] IETF RFC 3640 (November 2003): "RTP Payload Format for Transport of MPEG-4 Elementary Streams", J. van der Meer, D. Mackie, V. Swaminathan, D. Singer, P. Gentric.

[42] IETF RFC 1952 (May 1996): "GZIP file format specification version 4.3", P. Deutsch.

[43] ITU-T Recommendation H.264 (01/2012): "Advanced video coding for generic audiovisual services" | ISO/IEC 14496-10:2010: "Information technology - Coding of audio-visual objects - Part 10: Advanced Video Coding".

[44] Void.

[45] ITU-T Recommendation H.263 (1998): "Video coding for low bit rate communication".

[46] ITU-T Recommendation H.263 - Annex X (04/01): "Annex X: Profiles and levels definition".

[47] 3GPP TS 26.234: "Transparent end-to-end streaming service; Protocols and codecs".

[48] 3GPP TS 26.071: "AMR speech codec; General description".

[49] 3GPP TS 26.090: "AMR speech codec; Transcoding functions".

[50] 3GPP TS 26.073: "AMR speech Codec; C-source code".

[51] 3GPP TS 26.104: "ANSI-C code for the floating-point Adaptive Multi-Rate (AMR) speech codec".

[52] 3GPP TS 26.171: "AMR speech codec, wideband; General description".

[53] 3GPP TS 26.190: "Mandatory Speech Codec speech processing functions AMR Wideband speech codec; Transcoding functions".

[54] 3GPP TS 26.173: "ANCI-C code for the Adaptive Multi Rate - Wideband (AMR-WB) speech codec".

[55] 3GPP TS 26.204: "ANSI-C code for the floating-point Adaptive Multi-Rate Wideband (AMR-WB) speech codec".

[56] Scalable Polyphony MIDI Specification Version 1.0, RP-34, MIDI Manufacturers Association, Los Angeles, CA, February 2002.

[57] Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP Version 1.0, RP-35, MIDI Manufacturers Association, Los Angeles, CA, February 2002.

[58] "Standard MIDI Files 1.0", RP-001, in "The Complete MIDI 1.0 Detailed Specification, Document Version 96.1", The MIDI Manufacturers Association, Los Angeles, CA, USA, February 1996.

[59] Mobile DLS, MMA specification v1.0, RP-41 Los Angeles, CA, USA. 2004.

[60] Mobile XMF Content Format Specification, MMA specification v1.0, RP-42, Los Angeles, CA, USA. 2004.

[61] ITU-T Recommendation T.81 (1992) | ISO/IEC 10918-1:1993: "Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines".

[62] C-Cube Microsystems (September 1992): "JPEG File Interchange Format", Version 1.02.

[63] CompuServe Incorporated (1987): "GIF Graphics Interchange Format: A Standard defining a mechanism for the storage and transmission of raster-based graphics information", Columbus, OH, USA. See at http://www.dcs.ed.ac.uk/home/mxr/gfx/2d/GIF87a.txt.

[64] CompuServe Incorporated (1990): "Graphics Interchange Format: Version 89a", Columbus, OH, USA.

[65] IETF RFC 2083 (March 1997): "PNG (Portable Networks Graphics) Specification Version 1.0", T. Boutell.

[66] W3C Working Draft 27 October 2004: "Scalable Vector Graphics (SVG) 1.2", http://www.w3.org/TR/2004/WD-SVG12-20041027/.

[67] W3C Working Draft 13 August 2004: "Mobile SVG Profile: SVG Tiny, Version 1.2", http://www.w3.org/TR/2004/WD-SVGMobile12-20040813/.

[68] Standard ECMA-327 (June 2001): "ECMAScript 3rd Edition Compact Profile".

[69] WAP Forum Specification (Octobeer 2001): "XHTML Mobile Profile", http://www.openmobilealliance.org/tech/affiliates/wap/wap-277-xhtmlmp-20011029-a.pdf.

[70] ISO/IEC 10646-1 (2000): "Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane".

[71] The Unicode Consortium: "The Unicode Standard", Version 3.0 Reading, MA, Addison-Wesley Developers Press, 2000, ISBN 0-201-61633-5.

[72] 3GPP TS 26.245: "Transparent end-to-end Packet switched Streaming Service (PSS); Timed text format".

[73] IETF RFC 4646: "Tags for the Identification of Languages".

[74] ISO 639: "Codes for the representation of names of languages".

[75] ISO 3166: “Codes for the representation of names of countries and their subdivisions”.

[76] Void.

[77] 3GPP TS 23.003: “Numbering, addressing and identification”

[78] IETF RFC 1305 (March 1992): "Network Time Protocol (Version 3) Specification, Implementation".

[79] OMA Push OTA Protocol (25-April-2001): WAP-235-PushOTA-20010425-a http://www.openmobilealliance.org/tech/affiliates/LicenseAgreement.asp?DocName=/wap/wap-235-pushota-20010425-a.pdf

[80] IETF RFC 3711 (March 2004): "The Secure Real-time Transport Protocol (SRTP)", M. Baugher, D. McGrew, M. Naslund, E. Carrara, K. Norrman.

[81] IETF STD065/RFC 3551: "RTP Profile for Audio and Video Conferences with Minimal Control", Schulzrinne H. and Casner S., July 2003.

[82] IETF RFC 4648: "The Base16, Base32, and Base64 Data Encodings", Josefsson S., Ed., October 2006.

[83] IETF RFC 3023: "XML Media Types", M. Murata, S. St.Laurent, D. Kohn, January 2001.

[84] IETF RFC 5905: "Network Time Protocol Version 4: Protocol and Algorithms Specification", D. Mills, June 2010.

[85] OMA OMNA Registered PUSH Application ID list http://www.openmobilealliance.org/tech/omna/omna-push-app-id.htm

[86] 3GPP TR 26.936: “Performance characterization of 3GPP audio codecs”.

[87] 3GPP TS 25.413: "UTRAN Iu interface Radio Access Network Application Part (RANAP) signalling".

[88] IETF RFC 2326: "Real Time Streaming Protocol (RTSP)", Schulzrinne H., Rao A. and Lanphier R., April 1998.

[89] 3GPP TS 26.142: "Dynamic and Interactive Multimedia Scene".

[90] "Service Guide for Mobile Broadcast Services", Open Mobile Alliance, OMA-TS-BCAST_ServiceGuide-V1_1, Candidate Version 1.1 –14 Sep 2010.

[91] IETF RFC 5053 (October 2007): “Raptor Forward Error Correction Scheme for Object Delivery”, M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer.

[92] IETF RFC 5285: "A General Mechanism for RTP Header Extensions", D. Singer, H. Desineni, July 2008.

[93] IETF RFC 4396: "RTP Payload Format for 3rd Generation Partnership Project (3GPP) Timed Text", Rey J. and Matsui Y., February 2006.

[94] OMA-ERELD-DM-V1_2-20070209-A: "Enabler Release Definition for OMA Device Management, Approved Version 1.2".

[95] 3GPP TS 26.430: "Timed Graphics".

[96] 3GPP TS 36.300: "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2".

[97] 3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification".

[98] 3GPP TS 26.247: "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

[99] 3GPP TS 29.061: " Interworking between the Public Land Mobile Network (PLMN) supporting packet based services and Packet Data Networks (PDN)".

[100] IETF RFC 6064: "SDP and RTSP Extensions Defined for 3GPP Packet-Switched Streaming Service and Multimedia Broadcast/Multicast Service", M. Westerlund, P. Frojdh, January 2011.

[101] void

[102] void

[103] void

[104] 3GPP TS 36.443: “Evolved Universal Terrestrial Radio Access Network (E-UTRAN); M2 Application Protocol (M2AP)”.

[105] 3GPP TS 36.101: "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception".

[106] IETF RFC6681, "Raptor FEC Schemes for FECFRAME," M.Watson, T. Stockhammer and M. Luby, August 2012.

[107] IETF RFC6363, "Forward Error Correction (FEC) Framework," M.Watson, A. Begen and V. Roca, October 2011.

[108] 3GPP TS 36.304: "User Equipment (UE) procedures in idle mode".

[109] “Mobile Location Protocol (MLP)”, Open Mobile Alliance, OMA-LIF-MLP-V3_1, Approved Version 3.1 – 20 Sep 2011.

[110] 3GPP TR 26.946: "Multimedia Broadcast/Multicast Service (MBMS); User service guidelines".

[111] Marine, A., Reynolds, J., and G. Malkin, "FYI on Questions and Answers - Answers to Commonly asked "New Internet User" Questions", FYI 4, RFC 1594, Mars 1994

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply:

Broadcast session: See 3GPP TS 22.146 [2].

Forward Error Correction (FEC): in the context of MBMS, a FEC mechanism is used at the application layer to allow MBMS receivers to recover lost SDUs

FLUTE channel: equivalent to an ALC/LCT channel
An ALC/LCT channel is defined by the combination of a sender and an address associated with the channel by the sender (RFC 3926 [9]).

Frame-packed stereoscopic 3D video: a video consisting of two views in which both views were packed into a single stream before compression.

Multicast joining: See 3GPP TS 22.146 [2].

Multicast session: See 3GPP TS 22.146 [2].

Multimedia Broadcast/Multicast Service (MBMS): See 3GPP TS 22.146 [2].

MBMS over a Single Frequency Network: See 3GPP TS 25.346 [5]

MBMS user services: MBMS User Service may use more than one Multimedia Broadcast/Multicast Service (bearer service) and more than one Broadcast and/or Multicast session

See 3GPP TS 22.246 [3].

MBMS user service discovery/announcement: user service discovery refers to methods for the UE to obtain the list of available MBMS user services along with information on the user service and the user service announcement refers to methods for the MBMS service provider to make the list of available MBMS user services along with information on the user service available to the UE

MBMS user service initiation: UE mechanisms to setup the reception of MBMS user service data
The initiation procedure takes place after the discovery of the MBMS user service

MBMS delivery method: mechanism used by a MBMS user service to deliver content
An MBMS delivery method uses MBMS bearers in delivering content and may make use of associated procedures.

MBMS download delivery method: delivery of discrete objects (e.g. files) by means of a MBMS download session

MBMS streaming delivery method: delivery of continuous media (e.g. real-time video) by means of a MBMS streaming session

MBMS download session: time, protocols and protocol state (i.e. parameters) which define sender and receiver configuration for the download of content files

MBMS streaming session: time, protocols and protocol state (i.e. parameters) which define sender and receiver configuration for the streaming of content

RTP Session: The RTP and RTCP traffic sent to a specific IP multicast address and port pair (one port each for RTP and RTCP) during the time period the session is specified to exist. An RTP session is used to transport a single media type (e.g. audio, video, or text). An RTP session may contain several different streams of RTP packets using different SSRCs.

Stereoscopic 3D video: a video bitstream consisting of two views.

Unicast Bearer Service: Synonymously used as the term “UMTS Bearer Services with interactive and/or streaming traffic classes”.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

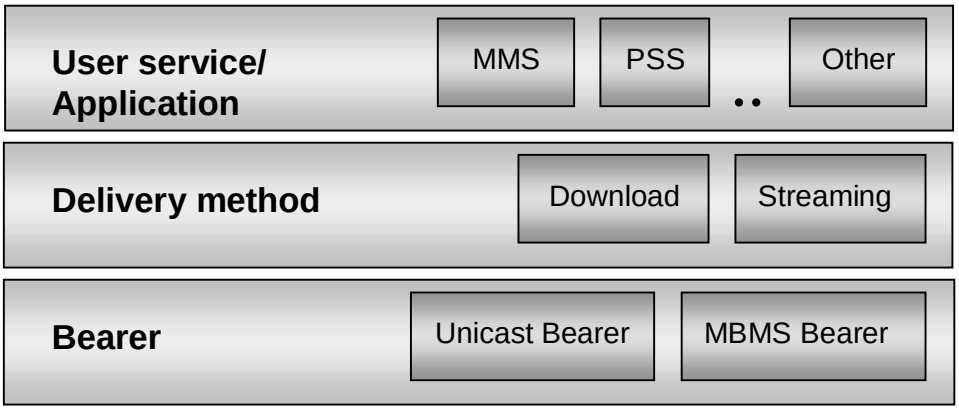
ALC	Asynchronous Layered Coding
APN	Access Point Name
AVC	Advanced Video Coding
BM-SC	Broadcast-Multicast - Service Centre

CC	Congestion Control
CPB	Coding Picture Buffer
DASH	Dynamic Adaptive Streaming over HTTP
DIMS	Dynamic and Interactive Multimedia Scenes
ERT	Expected Residual Time
ESI	Encoding Symbol ID
FDT	File Delivery Table
FEC	Forward Error Correction
FLUTE	File deLivery over Unidirectional Transport
FMT	Feedback Message Type
FQDN	Fully Qualified Domain Name
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
HDTV	High-DefinitionTeleVision
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
LCT	Layered Coding Transport
LI	Leap Indicator
MBMS	Multimedia Broadcast/Multicast Service
MBSFN	MBMS over a Single Frequency Network
MIME	Multipurpose Internet Mail Extensions
MMS	Multimedia Messaging Service
MPD	Media Presentation Description
MS	Mobile Station
MSK	MBMS Service Key
MTK	MBMS Traffic Key
MUK	MBMS User Key
NTP	Network Time Protocol
PDP	Packet Data Protocol
PSS	Packet-switched Streaming Service
PTM	Point To Multipoint
PTP	Point To Point
RTP	Real-time Transport Protocol
SBN	Source Block Number
SCT	Sender Current Time
SDP	Session Description Protocol
SDU	Service Data Unit
SEI	Supplemental Enhancement Information
SGSN	Serving GPRS Support Node
SNTP	Simple Network Time Protocol
TMGI	Temporary Mobile Group Identity
TOI	Transport Object Identifier
TSI	Transport Session Identifier
UDP	User Datagram Protocol
UE	User Equipment
UML	Unified Modelling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USD	User Service Description
UTC	Universal Time Coordinated
VCL	Video Coding Layer
XML	eXtensible Markup Language

4 MBMS system description

4.1 MBMS functional layers

Three distinct functional layers are defined for the delivery of MBMS-based service. They are Bearers, Delivery method and User service. Figure 1 depicts these layers with examples of bearer types, delivery methods and applications.



Bearers: Bearers provide the mechanism by which IP data is transported. MBMS bearers as defined in 3GPP TS 23.246 [4] and 3GPP TS 22.146 [3] are used to transport multicast and broadcast traffic in an efficient one-to-many manner and are the foundation of MBMS-based services. MBMS bearers may be used jointly with unicast PDP contexts in offering complete service capabilities.

Delivery Method: When delivering MBMS content to a receiving application one or more delivery methods are used. The delivery layer provides functionality such as security and key distribution, reliability control by means of forward-error-correction techniques and associated delivery procedures such as file-repair, delivery verification. Two delivery methods are defined, namely download and streaming. Delivery methods may be added beyond the current release. Delivery methods may use MBMS bearers and may make use of point-to-point bearers through a set of MBMS associated procedures.

User service: The MBMS User service enables applications. Different applications impose different requirements when delivering content to MBMS subscribers and may use different MBMS delivery methods. As an example a messaging application such as MMS would use the download delivery method while a streaming application such as PSS would use the streaming delivery method.

Figure 1: Functional Layers for MBMS User Service

4.2 MBMS User Service Entities

Figure 2 shows the MBMS user service entities and their inter-relations. Relation cardinality is depicted as well.

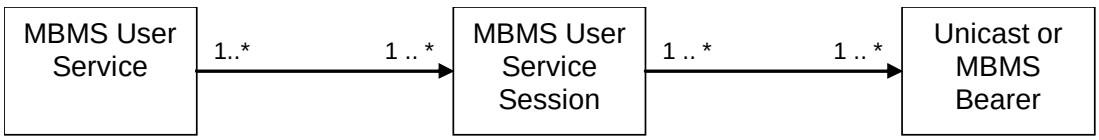


Figure 2: Entities and Relations

An MBMS user service is an entity that is used in presenting a complete service offering to the end-user and allowing him to activate or deactivate the service. It is typically associated with short descriptive material presented to the end-user, which would potentially be used by the user to decide whether and when to activate the offered service.

A single service entity can contain multiple distinct multimedia objects or streams, which may need to be provided over various MBMS download or MBMS streaming sessions. A download session or a streaming session is associated with either an unicast bearer or one or more MBMS bearers and a set of delivery method parameters specifying how content is to be received on the mobile side. The MBMS User Service Session may be mapped either on MBMS Bearer Services or on unicast bearer services.

A set of one or more MBMS bearers can be used for delivering data as part of an MBMS download or streaming session. As an example, the audio and visual parts can be carried on separate MBMS bearers. However, it is recommendated to transfer MBMS download and/or streaming sessions, which belong to the same MBMS user service on the same MBMS bearer service.

An MBMS bearer service (identified by TMGI) may be used to transport data for one or more MBMS download or streaming sessions (3GPP TS 22.246 [3], clause 5).

4.3 MBMS bearer service architecture

The MBMS Bearer Service Architecture is defined in 3GPP TS 23.246 [4]. The MBMS User Service interfaces to the MBMS system via 3 entities.

- The BM-SC.
- The GGSN (for GPRS) or MBMS-GW (for EPS).
- The UE.

The BM-SC provides functions for MBMS user service provisioning and delivery to the content provider. It can also serve as an entry point for IP MBMS data traffic from the MBMS User Service source.

The GGSN (for GPRS) or MBMS-GW (for EPS) serves as an entry point for IP multicast traffic as MBMS data from the BM-SC.

4.4 Functional Entities to support MBMS User Services

4.4.1 MBMS User Service Architecture

Figure 3 depicts the MBMS network architecture showing MBMS related entities involved in providing MBMS user services.

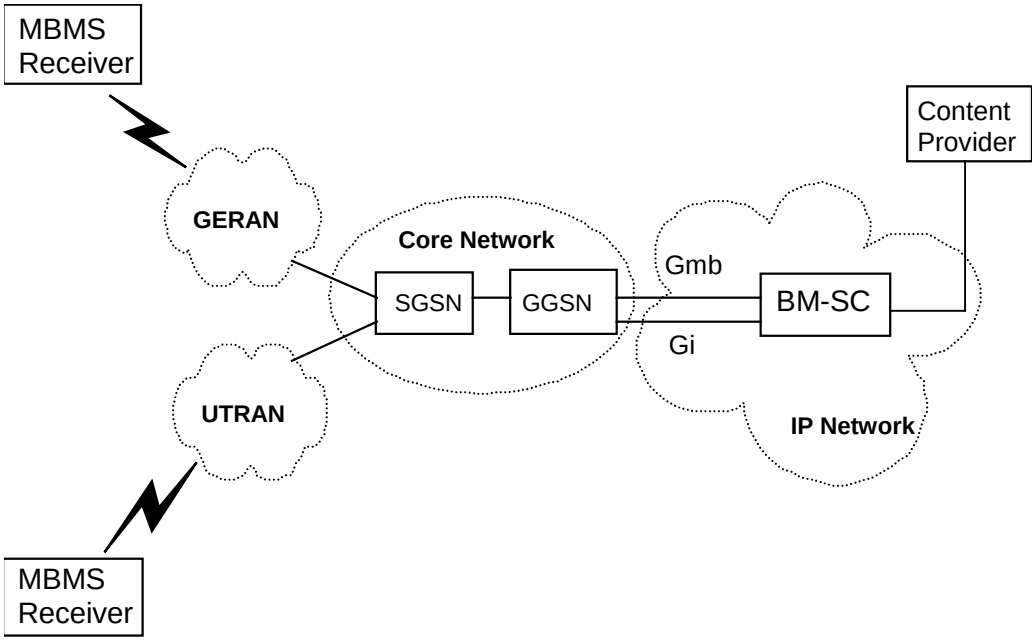


Figure 3: MBMS network architecture model for GPRS

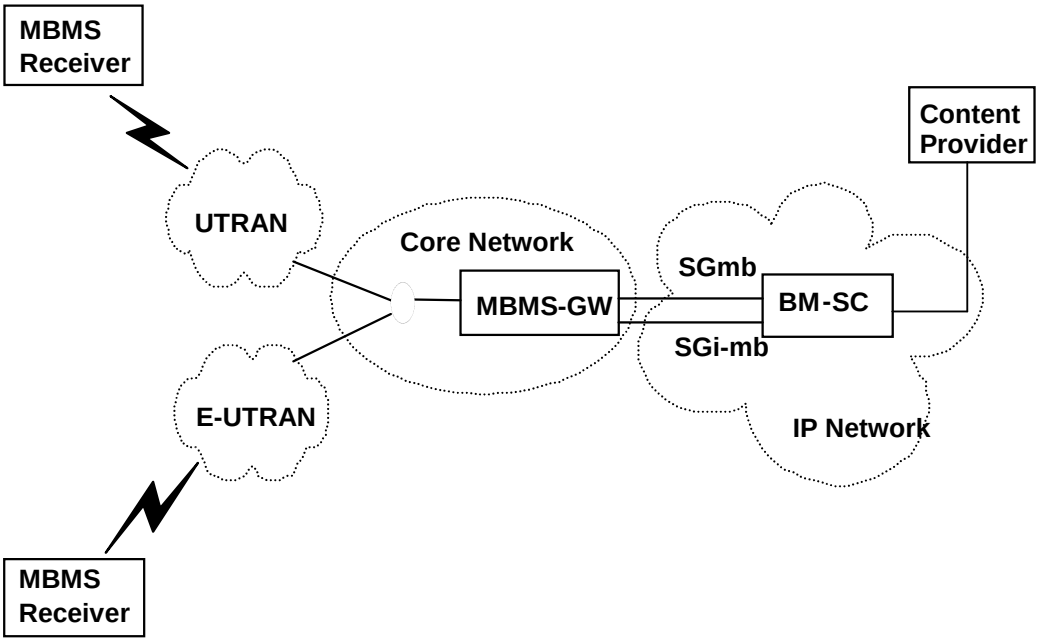


Figure 3b: MBMS network architecture model for EPS

MBMS User Service architecture is based on an MBMS receiver on the UE side and a BM-SC on the network side.

The use of the Gmb / SGmb and Gi / SGi-mb interface in providing IP multicast traffic and managing MBMS bearer sessions is described in detail in 3GPP TS 23.246 [4].

Details about the BM-SC functional entities are given in figure 4.

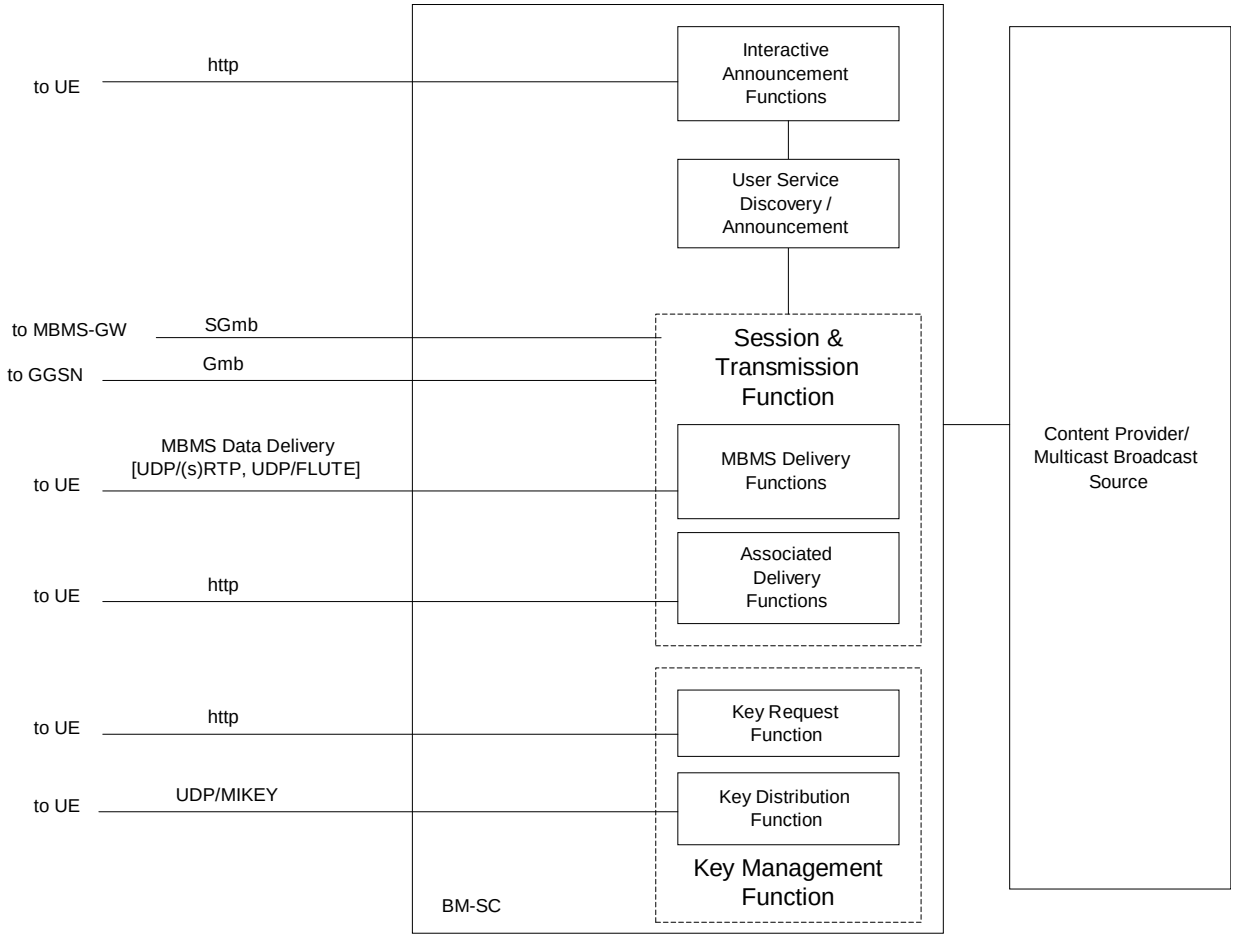


Figure 4: BM-SC sub-functional structure

The Session and Transmission function is further subdivided into the MBMS Delivery functions and the Associated Delivery functions.

The BM-SC and UE may exchange service and content related information either over point-to-point bearers or MBMS bearers whichever is suitable. To that end the following MBMS procedures are provided:

- User Service Discovery / Announcement providing service description material to be presented to the end-user as well as application parameters used in providing service content to the end-user.
- MBMS-based delivery of data/content from the BM-SC to the UE over IP multicast or over IP unicast.
  - The data/content is optionally confidentiality and/or integrity protected
  - The data/content is optionally protected by an forward error correction code
- Key Request and Registration procedure for receiving keys and key updates.
- Key distribution procedures whereby the BM-SC distributes key material required to access service data and delivered content.
- Associated Delivery functions are invoked by the UE in relation to the MBMS data transmission. The following associated delivery functions are available:
  - File repair for download delivery method used to complement missing data.
  - Delivery verification and reception statistics collection procedures.

The interfaces between internal BM-SC functions are outside the scope of the present document.

A "Proxy and Transport function" may be located between the "Session and Transmission Function" and the GGSN (for GPRS) or MBMS-GW (for EPS). The "Proxy and Transport function" is transparent to the "Session and Transmission function". The “Proxy and Transport” function is defined in sub-clause 5.1.3 of [4].

4.4.1a Content Provider / Multicast Broadcast Source

The Content Provider/Multicast Broadcast Source may provide discrete and continuous media, as well as service descriptions and control data, to the BM-SC to offer services at a time. An MBMS User Service may use one or several MBMS delivery methods simultaneously. The Content Provider/Multicast Broadcast Source may also be a 3rd Party Content Provider/Multicast Broadcast Source.

The Content Provider/Multicast Broadcast Source function may reside within the operator's network or may be provided from outside the operator's network. The Content Provider/Multicast Broadcast Source can also configure the Session and Transmission functions (e.g. delivery or associated delivery). The interface between the Content Provider/Multicast Broadcast Source and the BM-SC is outside the scope of the present document.

4.4.2 MBMS Key Management Function

The MBMS Key Management function is used for distributing MBMS keys (Key Distribution subfunction) to authorized UEs. Before the UE can receive MBMS keys, the UE needs to register to the Key Request subfunction of the Key Management function by indicating the MBMS User Service Id. Once registered, the UE can request missing MBMS keys from the BM-SC by indicating the specific MBMS key Id. In order for the UE to stop the BM-SC to send MBMS key updates a deregistration with the MBMS User Service Id is needed.

If the MBMS User Service does not require any MBMS data protection, then the UE shall not register for key management purposes.

A detailed description of all key management procedures is provided in 3GPP TS 33.246 [20].

4.4.3 MBMS Session and Transmission Function

The MBMS Session and Transmission function transfers the actual MBMS session data to the group of MBMS UEs using either MBMS Bearer Services or unicast bearer services. The MBMS Session and Transmission function interacts with the GGSN (for GPRS) through the Gmb Proxy function to activate and release the MBMS transmission resources. The MBMS Session and Transmission function interacts with the MBMS-GW (for EPS) through the SGmb Proxy function to activate and release the MBMS transmission resources.

The session and transmission function may compress headers of MBMS data in some cases. Further, the session and transmission function may need to add synchronization information for the MBMS payload e.g. in case of MBSFN transmissions. For details on usage of synchronization and header compression see 3GPP TS 23.246 [4] and 3GPP TS 25.346 [5].

The function contains the MBMS delivery methods, which use the MBMS bearer service for distribution of content. Further this function contains a set of Associated-Delivery Functions, which may be invoked by the UE in relation to the MBMS data transmission (e.g. after the MBMS data transmission).

The BM-SC Session and Transmission function is further described in later clauses of the present document as well as in 3GPP TS 23.246 [4].

MBMS user services data may be integrity and/or confidentiality protected as specified within 3GPP TS 33.246 [20], and protection is applied between the BM-SC and the UE. This data protection is based on symmetric keys, which are shared between the BM-SC and the UEs accessing the service.

MBMS user services may also be protected against packet loss between BM-SC and UE using a forward error correction code.

4.4.4 User Service Discovery / Announcement function

The User Service Discovery / Announcement provides service description information, which may be delivered via the Session and Transmission function or via the Interactive Announcement function. This includes information, which is

necessary to initiate an MBMS user service as described in sub-clause 5.3.1. Metadata for the service descriptions are described in sub-clause 5.2.

4.4.5 Interactive Announcement Function

An Interactive Announcement Function may offer alternative means to provide service descriptions to the UE using HTTP or be distributed through other interactive transport methods.

4.4.6 MBMS UE

The MBMS UE hosts the MBMS User Services receiver function. The MBMS receiver function may receive data from MBMS bearer services or from unicast bearer services. The MBMS receiver function may receive data from several MBMS User Services simultaneously. According to the MBMS UE capabilities, some MBMS UEs may be able to receive data belonging to one MBMS User Service from several MBMS Bearer Services simultaneously. The MBMS receiver function uses interactive bearers for user service initiation / termination, user service discovery and associated delivery procedures.

In case the MBMS user service is secured, the UE needs one or more cryptographic MBMS service keys, therefore the UE requests the relevant cryptographic MBMS service keys using the BM-SC Key Request function. The received keys (i.e. MSK) are then used for securing the MBMS session.

4.5 Usage of identity of MBMS session

The Session Identity of the MBMS session is provided with the MBMS session start procedure from the BM-SC to the GGSN (for GPRS) or MBMS-GW (for EPS) via the Gmb (for GPRS) or SGmb (for EPS) protocol in the MBMS Session Identity information element. The “MBMS Session Identity” information element is specified in [99]. The size of the Session Identity field is 1 octet. The MBMS Session Identity is forwarded with the MBMS SESSION START REQUEST message through the system and received by the MBMS UE with the paging message.

The usage of the MBMS Session Identity is optional. The MBMS Session Identity is only applicable to MBMS download delivery sessions. The MBMS transmission resources are activated as described in sub-clause 5.4. Each MBMS session of the MBMS User Service may be activated using a different MBMS Session Identity. The MBMS UE determines, based on the MBMS Session Identity value, whether the files of the upcoming MBMS download session were already received. If the files have already been completely received, the MBMS UE does not respond to the notification of the MBMS Session.

The association of MBMS Session Identities to files is determined by the BM-SC and communicated within the File Delivery Table. This association of a MBMS Session Identity to files is valid until a particular expiry time, also signalled within the File Delivery Table. If a UE has not received a File Delivery Table associating a given MBMS Session Identity to a specific file or set of files, or a previously received association has expired, then the UE shall assume that the MBMS Session Identity value is associated to new files which has not yet been received and shall respond as normal to MBMS notifications with that Session Identity value.

A single MBMS Session Identity value may be associated with a single file or with a set of files. Once a MBMS Session Identity value has been associated with a particular file or a set of files, this association shall not be changed before the expiry of the validity time for that MBMS Session Identity value. In particular, a File Delivery Table including some files that has previously been associated with a particular Session Identity value must include all files previously associated with that value, even if it is not intended to include all the files within the MBMS transmission session.

An FDT instance includes the MBMS Session Identity expiry time and associates the MBMS Session Identity expiry times with particular MBMS Session Identity values.

If the MBMS Session Identity is used by the BM-SC, the BM-SC shall also provide the session repetition number of that MBMS transmission session on the Gmb (for GPRS) or SGmb (for EPS) interface.

If the BM-SC starts using the MBMS Session Identity for one MBMS Bearer Service, the BM-SC may still decide not to use the MBMS Session Identity for a later MBMS transmission on that MBMS bearer service (e.g. when an MBMS session is transmitted only once).

After determining that all files for a MBMS Session Identity value has been received, the UE shall not respond to MBMS notifications for the MBMS Bearer Service with that MBMS Session Identity value until the MBMS Session Identity is expired. Once the MBMS Session Identity has expired, the content is no longer guaranteed to be repeated, and therefore the UE may begin to respond to MBMS notifications for the associated MBMS Bearer Service.

The BM-SC may send FDT instances on a separate transmission session or interleaved with other data packets of the same transmission session. An FDT instance may describe more files than the files to be transmitted over the same transmission session as that FDT instance.

4.6 Time Synchronization between the BM-SC and MBMS UEs

A number of MBMS metadata fragments and File Delivery Table (FDT) contain NTP encoded time values. NTP uses UTC as reference time and is independent from time zones. In order to process the time information from the BM-SC correctly, the MBMS UEs shall be time synchronized with the BM-SC with a tolerance of +/- 1 second. The BM-SC shall offer an SNTP [84] time server. The MBMS UEs should use SNTP to synchronize the time with the BM-SC. It is expected that the MBMS UE periodically requests SNTP time synchronization in order to keep the +/- 1 second tolerance. However, the MBMS UE should use the SNTP time synchronization service only as necessary to keep +/- 1 second accuracy, and should at most use the SNTP time synschonization once every 24 hours to avoid scalability issues.

To further prevent scalability issues, the MBMS UE should randomize its periodic SNTP requests over 1 hour just preceeding its determined SNTP request time.

SNTP time synchronization may be achieved either by using SNTP anycast [84], or SNTP unicast [84], depending on network support.

For network deployment where intermediate router nodes between the UE and BM-SC have anycast enabled, the BM-SC shall support the SNTP anycast mode. The MBMS UE sends a request to a designated IPv4 or IPv6 local broadcast address or multicast group address. One or more SNTP anycast servers reply and include a timestamp with their current time and its precision. BM-SC SNTP servers shall only respond if they have a valid synchronization time and shall not leave the timestamp blank, such that the SNTP Leap Indicator (LI) field shall not use the value 3 (warning: unsynchronised). The MBMS UE does not need to keep server address state data and changes in the SNTP server addressing will not affect each subsequent synchronization operation.

For IPv4, the Internet Assigned Numbers Authority (IANA) has assigned the multicast group address 224.0.1.1 for NTP, which is used by both multicast servers and anycast clients. For IPv6, the IANA has assigned the multicast group address FF0X:0:0:0:0:0:0:101. These NTP assignments apply to SNTP usage as well. The SNTP server will join these IP multicast groups.

For network deployment where intermediate router nodes between the UE and the SNTP servers do not have anycast enabled, the SNTP server(s) shall support unicast mode. The MBMS UE sends a request to its pre-configured SNTP server address FQDN [111]. The network may distribute the SNTP request traffic load to a pool of SNTP servers in the network, as long as the UE pre-configured SNTP FQDN address is unchanged. The way the network performs this load distribution is out of scope of this specification. SNTP servers shall only respond if they have a valid synchronization time and shall not leave the timestamp blank, such that the SNTP Leap Indicator (LI) field shall not use the value 3 (warning: unsychronized).

An MBMS UE shall select the SNTP mode to use as follows:

1. Attempt time synchronization using SNTP anycast;
2. If SNTP anycast procedure is successful then the UE should use SNTP anycast and continue using anycast for future periodic SNTP time synchronization over the same access network;
3. If the SNTP anycast procedure fails then it should use SNTP unicast and continue using unicast for future periodic SNTP time synchronization over the same access network.
4. In case of access network change detected by the UE, the UE should go to step 1 for its next periodic SNTP time synchronization.

5 Procedures and protocols

5.1 Introduction

This clause specifies the MBMS User service procedures and protocols.

5.2 User Service Discovery/Announcement

5.2.1 Introduction

User service discovery refers to methods for the UE to obtain a list of available MBMS user services or user service bundles along with information on the user services. Part of the information may be presented to the user to enable service selection.

User service announcement refers to methods for the MBMS service provider to announce the list of available MBMS user services and user service bundles, along with information on the user services, to the UE.

In order for the user to be able to initiate a particular service, the UE needs certain metadata information. The required metadata information is described in sub-clause 5.2.2.

According to 3GPP TS 23.246 [4], in order for this information to be available to the UE operators/service providers may consider several service discovery mechanisms. User service announcement may be performed over a MBMS bearer or via other means. The download delivery method is used for the user service announcement over a MBMS bearer. The user service announcement mechanism based on the download delivery method is described in sub-clause 5.2.3. The user service announcement using interactive announcement function is described in sub-clause 5.2.4. Other user service announcement and discovery mechanisms by other means than the download delivery method are out of scope of the present document.

5.2.2 MBMS User Service Description Metadata Fragments

5.2.2.1 Introduction

MBMS User Service Discovery/ Announcement is needed in order to advertise MBMS Streaming and MBMS Download User Services and User Service Bundles in advance of, and potentially during, the User Service sessions described. The User Services are described by metadata (objects/files) delivered using the download delivery method as defined in clause 7 or using interactive announcement functions.

MBMS User Service Discovery/Announcement involves the delivery of fragments of metadata to many receivers in a suitable manner. The metadata itself describes details of services. A *metadata fragment* is a single uniquely identifiable block of metadata. An obvious example of a metadata fragment would be a single SDP file ([14]).

The metadata consists of:

- a metadata fragment object describing details of a single or a bundle of MBMS user services (defined in sub-clause 11.2);
- a metadata fragment object(s) describing details of MBMS user service sessions (defined in sub-clause 7.3 and 8.3);
- a metadata fragment object(s) describing details of Associated delivery methods (defined in sub-clause 9.5);
- a metadata fragment object(s) describing details of service protection (defined in sub-clause 11.3);
- a metadata fragment object describing details of the FEC repair data stream;
- a metadata fragment object providing a Media Presentation Description (defined in sub-clause 11.2.1.2);
- a metadata fragment object(s) providing Initialisation Segments (defined in sub-clause 11.2.1.2);

- a metadata fragment object(s) providing a Schedule information description (defined in sub-clause 11.2A);
- a metadata fragment objects(s) providing filtering data for an MBMS User Service within a service bundle at the level of individual sessions of a given user service, or individual file contents within a user service (defined in sub-clause 11.2B). Only location filter data is defined in this release of the specification.

Metadata management information consists of:

- a metadata envelope object(s) allowing the identification, versioning, update and temporal validity of metadata fragment objects (defined in sub-clause 11.1).

A metadata envelope may have multiple metadata envelope instances. The metadata envelope and metadata fragment objects are transported as file objects in the same download session either as separate referencing files or as a single embedding file - see sub-clause 5.2.3.3. A single metadata envelope instance shall describe a single metadata fragment, and thus instances of the two are paired. A service announcement sender shall make a metadata envelope instance available for each metadata fragment instance. The creation and use of both an embedded envelope instance and a referenced envelope instance for a particular fragment instance is not recommended.

The metadata envelope and metadata fragment objects may be compressed using the generic GZip algorithm specified in RFC 1952 [42] as content/transport encoding for transmission. Where used over an MBMS bearer, this shall be according to Download delivery content encoding using FLUTE - see sub-clause 7.2.5.

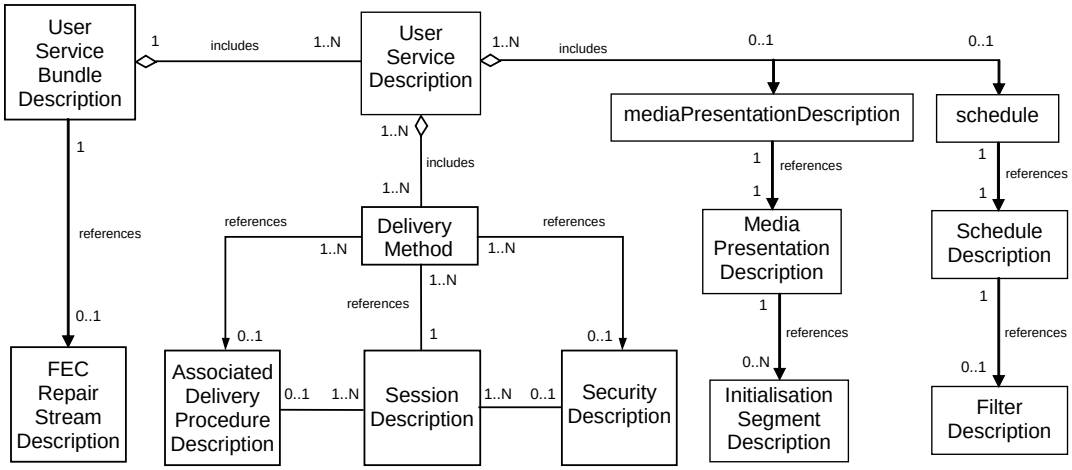


Figure 5: Simple Description Data Model

Figure 5 illustrates the simple data model relation between these description instances using UML [21] for a single User Service Bundle Description.

NOTE: "N" means any number in each instance.

One MBMS User Service Bundle Description shall contain one or more User Service Description instances. The User Service Bundle Description may refer to a single instance of the FEC Repair Stream Description metadata fragment.

In the event a MBMS User Service carries DASH-formatted contents, the *userServiceDescription* element, representative of the user service description, shall contain a *mediaPresentationDescription* element. The *mediaPresentationDescription* element shall in turn contain a reference to the Media Presentation Description metadata fragment whose data structure is identical to the MPD (Media Presentation Description) as defined in [98]. Furthermore, the Media Presentation Description fragment may refer to one or more Initialisation Segment metadata fragments whose data structure is identical to the Initialization Segment as defined in [98]. Also, when DASH-formatted contents are delivered by MBMS, at least one of the delivery methods shall be the download delivery method.

One MBMS User Service Description instance shall include at least one delivery method description instance. The delivery method description shall refer to one Session Description fragment.

The delivery method description may contain a reference to a Security Description fragment and an Associated Delivery Procedure fragment. Several delivery methods may reference the same Security Description fragment.

If the Associated Delivery Procedure fragment is present in the User Service Description instance, it may be referenced by one or more delivery methods.

An MBMS User Service Description allows the association of delivery methods to one or more access systems. The association is used to describe the use of separate access systems for the same MBMS User Service. One delivery method may be offered throughout one or more radio access systems. The use of separate MBMS bearer services for the same MBMS User Service is described in sub-clause 5.1.5.2 of [4].

One MBMS User Service Description instance may include at most one schedule instance. If included, the schedule instance shall refer to one Schedule Description fragment, and the UE can expect to receive MBMS User Service data during the time periods described in the Schedule Description fragment. In the case of a file download service, the Schedue Description fragment may include a file transmission schedule for file objects associated with the User Service. The UE may select which files to receive based on the file transmission schedule information in the Schedule Description fragment.

It is also possible for multiple User Service Description instances to reference the same Schedule Description fragment. In this case, the associated delivery schedule information shall include the file transmission schedule for files belonging to each of these User Services.

The Schedule Description may contain a reference to one Filter Description fragment, in which case the MBMS User Service is associated with filtering data which enables the UE to perform selective or targeted reception at either the session or the content file level of the User Service.

Multipart MIME [37] may be used to concatenate the descriptions into one document for transport.

5.2.2.2 Session Description

One or more session descriptions are contained in one session description object. The session description instance shall be formatted according to the Session Description Protocol (SDP) [14]. Each session description instance must describe either one Streaming session or one FLUTE Download session. A session description for a Streaming session may include multiple media descriptions for RTP sessions. The *sessionDescriptionURI* references the session description object. The session description is specified in sub-clause 7.3 for the MBMS download delivery method and in subclause 8.3 for the MBMS streaming delivery method.

5.2.2.3 Associated Delivery Procedure Description

The description and configuration of associated delivery procedures is specified in clause 9. The *associatedProcedureDescriptionURI* references the associated delivery procedure instance.

An associated delivery procedure description may be delivered on a dedicated announcement channel and updated on a dedicated announcement channel as well as in-band with an MBMS download session.

If an associated delivery procedure description for File-Repair operations is available, then the MBMS receiver may use the file repair service as specified in sub-clause 9.3.

If an associated delivery procedure description for reception reporting is available, then the MBMS receiver shall provide reception reports as specified in sub-clause 9.4.

5.2.2.4 Service Protection Description

The security description fragment contains the key identifiers and procedure descriptions for one delivery method. When different delivery methods use the same security description, the same security description document is referenced from the different delivery method elements.

The security description contains key identifiers and the server address to request the actual key material. To avoid overload situations, the same load balancing principles as in the associated delivery procedures are used. The key management server shall be selected as defined in sub-clause 9.3.5. The back-off time shall be determined as defined in sub-clause 9.3.4.

The XML schema for the service protection description is defined in sub-clause 11.3.

5.2.2.5 FEC Repair Stream Description

The streaming delivery method's FEC has separate stream for repair data, which is described by the FEC Repair Stream Description. The FEC Repair Stream Description shall be done using SDP [14]. This SDP file is referenced by the *bundleDescription* element in the service description. The FEC Repair Stream described is common for all FEC protected packet flows within the MBMS User Service Bundle Description instance.

5.2.2.6 Media Presentation Description

The Media Presentation Description fragment shall be a Media Presentation Description as specified in [98], containing descriptive information on the media presentation. This information will be used by the DASH client to construct the associated media presentation as a streaming service to the end user.

Availability of this metadata fragment is indicated by the presence of the *mediaPresentationDescription* element in the MBMS User Service Description fragment. In that case, at least one of the delivery methods shall be a download delivery method. The actual URI to the Media Presentation Description fragment is provided by the element *mpdURI* in the *mediaPresentationDescription* element.

5.2.2.7 Schedule Description

The Schedule Description metadata fragment is specified in sub-clause 11.2A.

The schedule description information may describe the schedule of an MBMS User Service in terms of

- start/stop lists, or
- reoccurring events
- The service ID or service Class to which the file schedule may apply,

The UE can expect to receive MBMS data during the described time periods when the session for the User Service is active.

The Schedule Description fragment may also include the schedule for when the files of a download delivery MBMS User Service are to be transmitted. The file schedule information is defined in terms of:

- The service ID or service Class to which the file schedule may apply,
- The list of file delivery schedule information consisting of:
  - A file URI to identify a given file being transmitted,
  - A list of broadcast delivery start and end times,

Note that such file schedule information would not be useful for download delivery services transporting DASH segments.

When including file delivery schedule, the schedule description fragment may capture the file transmission schedule for multiple User Services.

The schedule information contains a schedule update time, allowing the UE to know when to update its current schedule.

A Schedule Description fragment may be delivered as a metadata fragment on the service announcement channel and may be updated in-band with an MBMS download session. When describing the file delivery schedule for multiple user services, the Schedule Description fragment may be carried on an MBMS download delivery session dedicated to the transport of file schedule information. The mechanism UEs use to discover this file delivery schedule session is outside the scope of this specification.

The Schedule Description fragment may reference a Filter Description fragment, in which case the MBMS User Service is associated with filtering information which enables selective/targeted reception at one of the two mutually-exclusive levels:

i. by individual sessions of the User Service, and

ii. by individual content files of the User Service.

Detailed description of the alternative means to establish association between the Schedule Description and Filter Description fragments, and related filtering semantics, are provided in sub-clauses 11.2A and 11.2B.

5.2.2.8 Filter Description

The Filter Description metadata fragment contains filter data to enable selective/targeted UE reception of MBMS User Services or contents. When present in the USD, as indicated by the reference from the Schedule Description, it supports mechanisms to filter services/contents for intended ("positive") reception, as previously described in sub-clause 5.2.2.7. The intended usage of the filter data is defined by the way in which the Filter Description is referenced by the Schedule Description. As an example, each session of a DASH-encoded streaming service sent via the MBMS download delivery method may be associated with a unique filtering criterion, to enable targeted reception by specific UEs of data carried in that session. As another example, one or more content file items belonging to a download delivery service may be affiliated with a specific filter data instance which defines the rules for intended download of those files.

One or more *filterData* elements may be present in the Filter Description and is uniquely identified by its *id* attribute. A given instance of the filter data may be applicable to more than one MBMS User Service – for example, it may be intended for multiple User Services belonging to a User Service bundle to receive the same filtering treatment specified by that filter data instance. In this release of the specification, when present, the Filter Description fragment shall strictly contain location filtering data as defined by the *locationFilter* element.

5.2.3 User Service Announcement over a MBMS bearer

5.2.3.1 General

Both the metadata envelope and metadata fragment objects are transported as file objects in the same download session.

To receive a Service Announcement User Service the client shall obtain the session parameters for the related MBMS download session transport. This may be achieved by pre-storing the related session parameters in the MBMS UE or via delivery over OMA PUSH[79].

NOTE: The user service announcements are not protected when sent over MBMS bearer. See 3GPP TS 33.246 [20]

5.2.3.2 Metadata Envelope Transport

The metadata envelope object is transported as a file object in the same MBMS service announcement download session as its metadata fragment file object (i.e. in-band with the metadata fragment session).

5.2.3.3 Metadata Envelope and Metadata Fragment Association with FLUTE

The MBMS Download service announcement session FDT Instances provide URIs for each transported object. The metadata envelope instance *metadataURI* field shall use the same URI for the metadata fragment as is used in the FDT Instances for that metadata fragment file. Thus, the fragment can be mapped to its associated envelope in-band of a single MBMS download session.

In the referencing case, each metadata envelope and corresponding metadata fragment shall be grouped together by the FDT using the grouping mechanism described in sub-clause 7.2.6. This reduces the complexity of requesting both fragment and envelope for each pair, thus it is recommended that only the metadata fragment (fileURI) be requested from the download client (which will result in both fragment and envelope being received using the grouping mechanism).

5.2.4 User Service Announcement using Interactive Announcement Function

User service descriptions may be transported to the UE using HTTP and other interactive transport methods. A BM-SC may provide the service descriptions on request. Further protocol specifications of interactive announcement functions are outside of the scope of this specification.

Aggregated MBMS service announcement documents as specified in sub-clause 5.2.5 may be used with the interactive announcement functions. UEs shall support the disassembly of aggregated MBMS service announcement documents. UEs shall support Gzip decoding of MBMS service description objects for interactive transport (BM-SC use of Gzip is optional in accordance with sub-clause 5.2.2).

The BM-SC may use Metadata Envelopes as described in clause 11.1, and UEs shall support their use with the Interactive Announcement Function. Where metadata envelopes are not used, only the latest delivery of a metadata fragment shall be used by the UE, and the BM-SC shall ensure timely, consistent, size-limited and secure delivery of metadata by means outside the scope of this document.

5.2.5 User Service Announcement over point-to-point push bearers.

5.2.5.1 General

User service announcement over point-to-point push bearers have several characteristics that differ from user service announcement over a MBMS bearer. It is not essential that the metadata envelope made available by the service announcement sender is transmitted to the MBMS terminal. In the case that both the metadata envelope and metadata fragment objects are transported, it is a limitation of the solution that the metadata fragment must either be embedded within the metadata envelope, or that the metadata fragment must be referenced by the metadata envelope and they are both contained within a multipart MIME container [37]. In either configuration, both the metadata envelope and the metadata fragment objects are transported as file objects in the same download session.

This sub-clause covers both metadata transport and metadata fragmentation aspects of Service Announcement. Service Announcement over point-to-point push bearers is specified.

NOTE: The user service announcements are not protected when sent over point-to-point push bearers. See 3GPP TS 33.246 [20].

5.2.5.2 Supported Metadata Syntaxes

The supported metadata syntaxes are as defined in sub-clause 11.1 of this document.

5.2.5.3 Consistency Control and Syntax Independence

The consistency control and syntax independence is as defined in sub-clause 11.1 of this document.

5.2.5.4 Metadata Envelope Definition

The metadata envelope definition is as defined in sub-clause 11.1 of this document.

5.2.5.5 Delivery of the Metadata Envelope

An instance of metadata fragment shall either be embedded within the metadata envelope or be included in a multipart MIME container together with the envelope. The envelope and fragment are, by definition, transported together and in-band of the same transport session.

The Metadata Envelope includes a reference (*metadataURI*) to the associated metadata fragment using the same URI as the fragment file is identified by in the Service Announcement. Thus, Metadata Envelope can be mapped to its associated metadata fragment.

5.2.5.6 Metadata Envelope Transport

The metadata envelope object is transported as a file object in the same MBMS service announcement download session as its metadata fragment file object (i.e. in-band with the metadata fragment session).

5.2.5.7 User service announcement over SMS bearers

User service announcements over SMS bearers are formatted according to the OMA Push OTA specification [79].

OTA-WSP shall be used over the SMS bearer. Application port addressing shall be used as specified in [79]. The application ID to be used is 0x9023 as allocated by OMNA [85].

Either confirmed or unconfirmed push may be used. In either case, the primitive shall contain the Push Headers parameter. Within this parameter, the Content-Type header shall be included and the Content-Encoding header shall be included if GZip is used.

5.2.5.8 User service announcement over HTTP push bearers

User service announcements over HTTP push bearers are formatted according to the OMA Push OTA specification [79].

OTA-HTTP shall be used over the HTTP push bearer. Application port addressing shall be used as specified in [79]. The application ID to be used is 0x9023 as allocated by OMNA [85].

The Content-Encoding header shall be included if GZip is used.

5.2.6 Metadata Fragment Encapsulation to aggregate Service Announcement Documents

The present document defines a number of metadata fragments to describe MBMS user services. A metadata fragment is a single uniquely identifiable block of metadata. Generally, more than one metadata fragment is necessary to provide all necessary parameters to initiate an MBMS User Service. Typically, metadata fragments are provided in separate documents. Each metadata fragment is labelled with its MIME type.

Multipart MIME may be used to encapsulate metadata fragments into an aggregate service announcement document. The aggregate document may contain metadata fragments of several MBMS user services. It is recommended, that any such aggregate service announcement document contains all the referenced metadata fragments of each MBMS user service description it contains (i.e. in the same multipart MIME structure).

An aggregate service announcement document shall encapsulate metadata fragments according to RFC 2557 [37]. The first encapsulated file of an aggregate service announcement document is the root resource. The root resource shall be either an MBMS user service description or a metadata envelope (as a referencing index). The service description metadata is described in sub-clause 5.2.2 and defined in sub-clause 11.2. The metadata envelope is defined in sub-clause 11.1.

The type field of the multipart/related header shall be set to application/mbms-user-service-description-parameter in case the root resource is a user service description instance. The type field of the multipart/related header shall be set to application/mbms-envelope in case the root resource is a metadata envelope.

5.2.7 Registration and Deregistration Procedure for MBMS User Service Consumption

The MBMS User Service Description Fragment may include a registration description. If the registration description is present in the MBMS User Service Description Fragment, then the UE shall use the registration and deregistration procedures as defined in this section.

A registration request is then initiated by the UE, in order to receive the complete user service description. The registration procedure is performed using HTTP 1.1 [18] POST message towards the indicated RegistrationURL.

A successful registration response shall start with a 200 OK status line in the response header and shall contain in the body the metadata fragments that are referenced by the USD in a multipart MIME container.

The registration request shall be formatted according to the following XML schema and using the RegistrationRequest element.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           xmlns=" urn:3GPP:metadata:2008:MBMS:Registration"
           elementFormDefault="qualified"
           targetNamespace=" urn:3GPP:metadata:2008:MBMS:Registration">

    <xs:element name="RegistrationOperationRequest">
        <xs:complexType>
            <xs:choice>
                <xs:element name="RegistrationRequest" type="RegistrationOperationRequestType"/>
                <xs:element name="DeregistrationRequest" type="RegistrationOperationRequestType"/>
            </xs:choice>
        </xs:complexType>
    </xs:element>

    <xs:complexType name="RegistrationOperationRequestType">
        <xs:attribute name="IMEI" type="xs:string" use="optional"/>
        <xs:attribute name="MSISDN" type="xs:string" use="required"/>
        <xs:attribute name="ServiceID" type="xs:anyURI" use="required"/>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>

</xs:schema>
```

A de-registration procedure is used by the UE to de-register at the end of the user service consumption, in case a registration procedure has been performed. The de-registration request shall be sent to a registration server (preferably the one with which the registration procedure has been performed). The de-registration procedure consists of sending an HTTP 1.1 POST request with an XML body formatted according to the XML schema above, using the DeregistrationRequest element.

The MIME type of the message body of the registration and deregistration request shall be set to “text/xml”.

The *IMEI* element contains, if present, the International Mobile Equirment Identifier as defined in [77].

The *MSISDN* element contains the Mobile Subscriber ISDN Number as defined in [77].

The *ServiceID* element contains the unique MBMS User Service Identifier as defined in clause 11.2.1.1.

5.3 User Service Initiation/Termination

5.3.1 Initiation of MBMS Bearer Service based Services

MBMS User Service initiation refers to UE mechanisms to set-up the reception of MBMS user service data. During the User Service Initiation procedure, a set of MBMS Bearers may be activated. The User Service Initiation procedure takes place after the discovery of the MBMS user service.

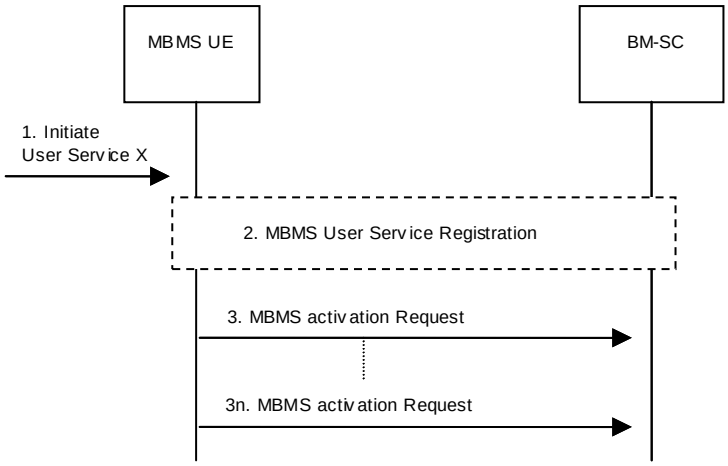


Figure 6: Initiation of an MBMS User Service

1. The User Service Initiation Procedure is triggered and takes a User Service Description as input that has been obtained e.g. by executing the MBMS User Service discovery and announcement functions.

2. The MBMS UE registers to the MBMS User Service, if registration is required for the MBMS User Service. If security functions are activated for the MBMS User Service, the MBMS UE requests MBMS service keys. The keys are sent to the UE, after the user is authorized to receive the MBMS service. The request shall be authenticated. Details on the MBMS User Service Registration procedure are described in 3GPP TS 33.246 [20].

3. The MBMS UE uses the MBMS activation procedure to activate the MBMS Bearer Service. The MBMS activation procedure is the MBMS Multicast Service activation procedure and the MBMS Broadcast activation procedure as defined in 3GPP TS 23.246 [4]. In case the MBMS Broadcast Mode is activated, there is no activation message sent from the UE to the BM-SC. The activation is locally in the UE. Note that the MBMS Bearer Services may already be active and in use by another MBMS User Service.

3n. In case the MBMS User Service uses several MBMS Bearer Services, the User Service Description contains several description items. In that case, the MBMS receiver function repeats the activation procedure for each MBMS Bearer Service as described in 3.

5.3.2 Termination of MBMS Bearer Service based Services

MBMS user service termination refers to the UE mechanisms to terminate the reception of MBMS user services. A set of MBMS Bearers may be deactivated during this procedure.

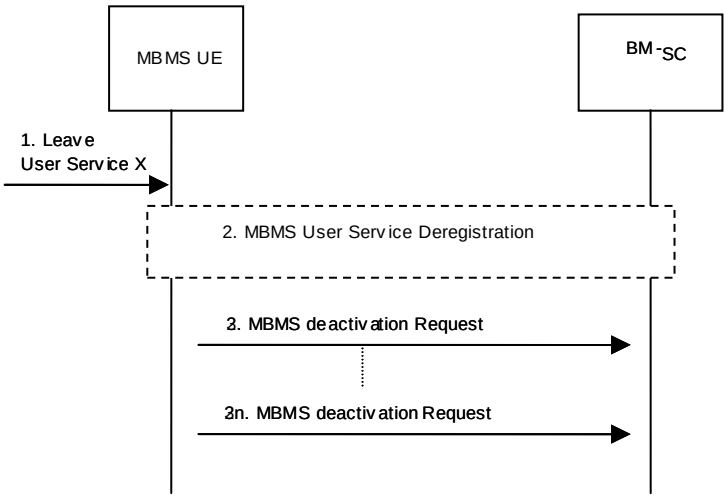


Figure 7: Termination of an MBMS user service

1. The User Service termination Procedure is triggered. A reference to the User Service to terminate is provided as parameter.

2. The MBMS UE deregisters, when registration was required for the MBMS User Service. If security functions are activated for the MBMS User Service, the MBMS UE deregisteres the security association for the MBMS User Services. Details on the MBMS User Service Deregistration procedure are described in 3GPP TS 33.246 [20].

3. If no other MBMS User Service uses the MBMS Bearer service, the MBMS UE uses the MBMS deactivation procedure to deactivate the MBMS Bearer Services. The MBMS deactivation procedure represents the MBMS Multicast service deactivation procedure and the MBMS Broadcast deactivation procedure as described in 3GPP TS 23.246 [4]. In case the MBMS Broadcast Mode is deactivated, there is no message sent to the BM-SC. The deactivation is only locally in the UE.

3n. In case the MBMS User Service uses several Bearer Services, the UE repeats the deactivation procedure for each Bearer Service as described in 3.

5.3.3 Initiation of Unicast Bearer Service based Services

Unicast Bearer Service based MBMS User Service initiation refers to the mechanisms to set-up the reception of MBMS user service data via a UMTS/EPS Bearer Service with interactive and/or streaming straffic class.

In case of the initiation of a MBMS Streaming delivery method or a combined MBMS Streaming and MBMS Download delivery method, the Packet Switched Streaming Service (PSS) as defined in 3GPP TS 26.234 [47] shall be used. The establishement of a PSS session is described in clause 5.1 of 3GPP TS 26.234 [47].

In case of the initiation of a MBMS Download delivery method, the MBMS UE is registered in the BM-SC for OMA-PUSH based reception of the files with the BM-SC.

5.3.4 Termination of Unicast Bearer Service based Services

Unicast Bearer Service based MBMS user service termination refers to the mechanisms to terminate the reception of MBMS user service data via a UMTS/EPS Bearer Service with interactive and/or streaming straffic class.

In case of the termination of a MBMS Streaming delivery method or a combined MBMS Streaming and MBMS Download delivery method, the Packet Switched Streaming Service (PSS) as defined in 3GPP TS 26.234 [47] shall be used. The termination of a PSS session is described in clause 5.3 of 3GPP TS 26.234 [47].

In case of the termination of a MBMS Download delivery method, the MBMS UE is deregistered in the BM-SC so that the OMA-PUSH based reception of the files with the BM-SC will be terminated.

5.3.5 Scalable Service Initiation and Termination for MBMS Services

5.3.5.1 General

MBMS service initiation and termination as defined in clauses 5.3.1 to 5.3.4 may consist of network interactions such as sending an IGMP Join or Leave message to the network as described in sections 8.2 and 8.7 of 3GPP TS 23.246 [4]. Initiation and termination procedures may be triggered at the MBMS UE by the user or be scheduled to happen automatically. Upon (or after) receiving a user service announcement, the MBMS UE may render the information about the advertised services to the user to assist him in the service selection. The user may decide to receive a given service and hence trigger the service initiation procedure. Alternatively, the user may declare his interest in a specific service a-priori and upon receiving the service announcement for that specific service, the MBMS UE may schedule the initiation procedure at or around the start time of the session. Similarly, the MBMS UE may schedule the termination procedure at or around the session end time.

As a consequence, MBMS UEs may be oriented to start their service initiation and termination procedures at the same time or during a relatively short period. This may cause network congestion, especially during the multicast of a popular service, as all MBMS UEs may be time synchronized.

5.3.5.2 Randomization of Service Initiation over Time

The MBMS User Service description may contain parameters to uniformly randomize the User Service Initiation procedures of the MBMS UEs. Security functions may be part of the User Service Initiation procedure as defined in clause 5.3.1. If a user service initiation randomization is defined for a user service, then the overload prevention definition in the service protection description shall be ignored for the service initiation. For randomizing the time of the initiation procedure, the MBMS UE shall understand the following parameters, which may be signalled by the BM-SC in the MBMS user service description as described in section 11.2.1:

1. *initiationStartTime* parameter is used by the BM-SC to signal to the MBMS UE the start time for the User Service Initiation procedure randomization period. If the *initiationStartTime* parameter is not present, the MBMS UE uses the time of the Service Annoucement reception as the start time.

2. *protectionPeriod* parameter is used by the BM-SC to signal to the MBMS UE the duration of the critical time periods, during which congestion shall be avoided. The MBMS UEs shall randomly spread the initiation procedure using the *randomTimePeriod* during this protection period.

3. *randomTimePeriod* parameter is used by the BM-SC to signal to the MBMS UE the duration of an interval over which initiation procedures shall be randomly deferred. The MBMS UE calculates a random time out of the *randomTimePeriod* interval to defer the execution of the initiation procedure.

The MBMS UE shall start its initiation procedure immediately if the procedure is triggered outside of protection periods.

5.3.5.3 Randomization of Service Termination over Time

The MBMS User Service description may contain parameters to uniformly randomize the User Service Termination procedures of the MBMS UEs. For randomizing the time of the termination procedure, the MBMS UE shall understand the following parameters, which may be signalled by the BM-SC in the MBMS USD as described in section 11.2.1:

1. *protectionPeriod* parameter is used by the BM-SC to signal to the MBMS UE the duration of the critical time period, during which congestion needs to be avoided. The MBMS UEs shall randomly spread the termination procedure using the *randomTimePeriod* during this period and starting from the session end time.

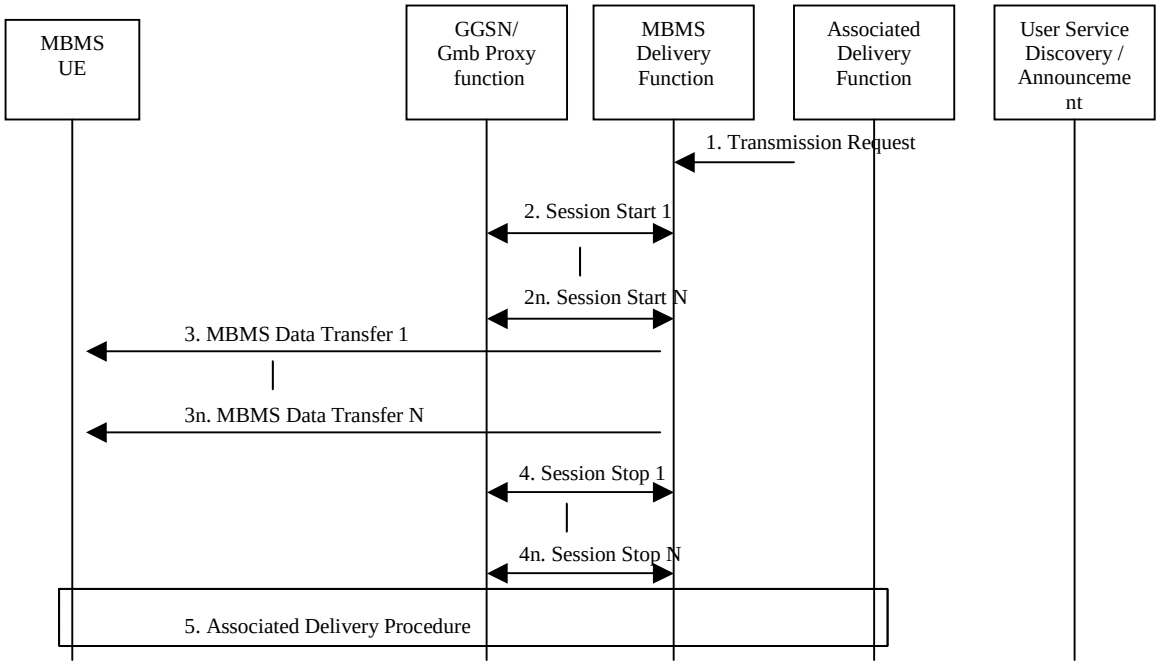
2. *randomTimePeriod* parameter is used by the BM-SC to signal to the MBMS UE the duration of an interval over which termination procedures shall be randomly deferred. The termination procedure is only randomized during the *protectionPeriod*.

If the termination procedure is triggered before the session end time or after the protection period end time, the MBMS UE shall start its termination procedure immediately. If it is in a protection period, the MBMS UE shall defer its termination procedure to a random time spread over an interval of duration *randomTimePeriod*.

5.4 MBMS Data Transfer Procedure

5.4.1 MBMS Data Transfer Procedure using MBMS Bearer Services

MBMS Data Transfer procedure using MBMS Bearer Services refers to the network (and UE) mechanism to transfer (and receive) data for one MBMS User Service on one or several MBMS Bearer Services.



NOTE: Security related interactions are not depicted in the sequence.

Figure 8: Procedure of MBMS Data Transfer

1. The MBMS Delivery Method for the MBMS User Service is triggered by the MBMS User Service Provider. Note, details of the trigger are beyond of the present document.

2. - 2n. The MBMS Delivery function uses the MBMS Session Start Procedure to the GGSN and/or MBMS-GW, possibly through the Gmb Proxy and/or the SGmb Proxy function to activate all MBMS Bearer Services, which belong to the MBMS User Service. The MBMS Bearer service to be activated is uniquely identified by the TMGI.

Note. MBMS Bearer services might be activated only to a subset of the available access systems (see 3GPP TS 23.246 [4]). In case MBMS User Services or delivery methods are not available throughout all access systems, the BM-SC describes this transmission strategy in the MBMS User Service Description (see sub-clause 5.2.2).

3. - 3n. The data of the MBMS user service are transmitted to all listening MBMS UEs. Several MBMS Bearer services may be used to transmit the MBMS user service data. MBMS user service data may be integrity and/or confidentiality protected. In case MBMS user service data are integrity and/or confidentiality protected, MBMS traffic keys are delivered simultaneously on the same or a different MBMS bearer. Optionally, syncronization information for MBSFN may be added to the MBMS User Data. The headers of MBMS User data may optionally be compressed (see 3GPP TS 23.246 [4] and TS 25.346 [5])

4. - 4n. The MBMS Delivery function uses the MBMS Session Stop procedure to trigger the GGSN and/or MBMS-GW, possibly through the Gmb and/or SGmb Proxy function to release all MBMS Bearer Service for this User Service. A unique identifier for the MBMS Bearer service to be deactivated (i.e. the TMGI) is passed on as a parameter.

5. In case associated delivery procedures are allowed or requested for an MBMS User Service, the MBMS UE sends an associated-delivery procedure request to the associated -delivery function. The BM-SC may authenticate the user. See 3GPP TS 33.246 [20]. The MBMS UE may need to wait a random time before it starts the associated delivery procedure according to clause 9.

5.4.2 MBMS Data Transfer Procedure using other UMTS Bearer Services

MBMS Data Transfer procedure using other UMTS Bearer Services refers to the network (and UE) mechanism to transfer (and receive) data for one MBMS User Service on one or more Unicast Bearer Services.

In case the MBMS Data belong to a MBMS Streaming delivery session or a combined MBMS Streaming and MBMS Download delivery session, the Packet Switched Streaming Service (PSS) as defined in 3GPP TS 26.234 [47] shall be used.

In case the MBMS Data belong to a MBMS Download delivery session, the MBMS data is transferred using OMA-PUSH.

5.5 MBMS Protocols

Figure 9 illustrates the protocol stack used by MBMS User services. The grey-shaded protocols and functions are outside of the scope of the present document. MBMS security functions and the usage of HTTP-digest and SRTP are defined in 3GPP TS 33.246 [20].

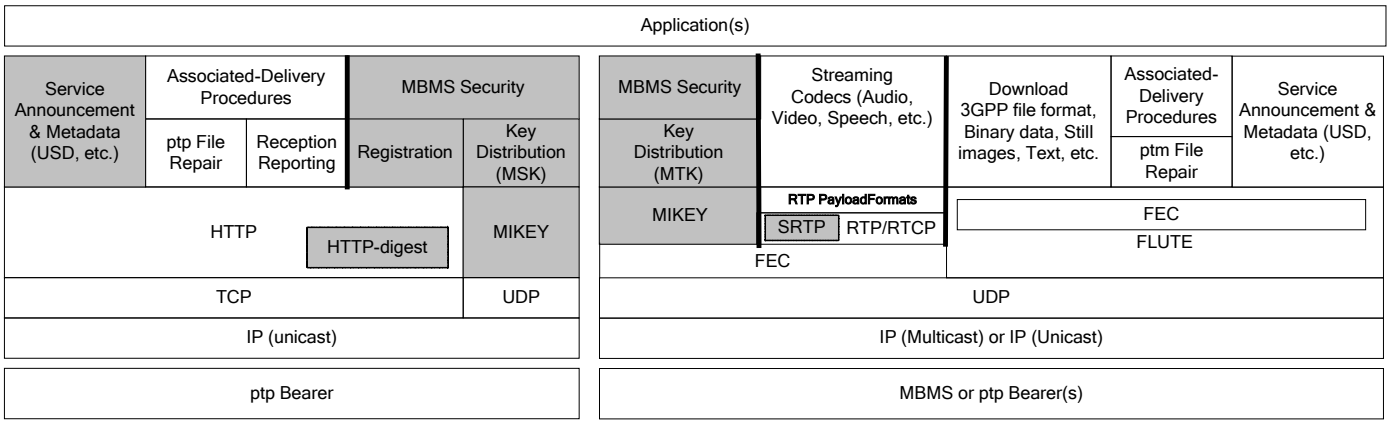


Figure 9: Protocol stack view of the MBMS User Services

5.6 3GP-DASH and MBMS

The 3GPP Dynamic Adaptive Streaming over HTTP (3GP-DASH) as defined in [98] specifies formats and methods that enable the delivery of streaming service(s) from standard HTTP servers to DASH client(s). It specifies the description of a collection of media segments and auxiliary metadata (all referenced by HTTP-URLs) through a Media Presentation Description (MPD).

MBMS is designed to serve large receive groups with same content. The MBMS Download Delivery Method is designed to deliver an arbitrary number of objects via MBMS to a large receiver population. MBMS Download defines several methods to increase reliability such as FEC and file repair. The download delivery method allows the delivery of DASH segments, Media Presentation Descriptions as well as other objects referenced in the MPD as defined in [98].

In order to support DASH Streaming in MBMS, the USD metadata fragment for a service shall contain a `mediaPresentationDescription` element referencing an MPD, which is also a metadata fragment describing the service. This is a reference to a Media Presentation Description metadata fragment as defined in [98]. If the USD contains a `mediaPresentationDescription` referencing an MPD, then

1. the user service shall be a download delivery service, i.e. shall include at least one `downloadDelivery` method element referencing an SDP that describes FLUTE transport.

2. The download delivery session shall deliver objects that are referenced by the MPD, all updates of the MPD and objects that are referenced by any update of the MPD.

3. If a Segment is delivered as a FLUTE object then all of the following shall hold:

a. The download delivery session shall deliver segments such that the last packet of the delivered object is available at the UE latest at its segment availability start time as announced in the MPD

b. The Content-Location element in the FDT for the delivered object shall match the Segment URL in the MPD.

4. If an MPD update is delivered as a FLUTE object then all of the following shall hold:

   a. The Content-Location element in the FDT for the delivered object shall match the `mpdURI` of the `mediaPresentationDescription` element.

   b. The MPD update shall be a valid update to a previously delivered MPD or an MPD delivered via the USD.

5. If any other resource in the MPD is delivered (e.g. xlinked resource, metrics, etc.) then

   a. The Content-Location element in the FDT for the delivered object shall match the URL of the object in the MPD.

   b. The download delivery session shall deliver objects such that the last packet of the delivered object is available at the UE latest at the earliest time a DASH client operating on the delivered MPD sequence may ask for the resource.

In the case a real-time streaming service is provided as DASH streaming over MBMS, then the **`MPD`**`@type` (attribute 'type' of the MPD) shall be set to "dynamic", i.e. this indicates that the segments get available over time, latest at its announced segment availability start time. When **`MPD`**`@minimumUpdatePeriod` (attribute 'minimumUpdatePeriod' of the MPD) is present, then the UE should expect MPD updates to be sent in the FLUTE session with the media segments and treat these updates as defined in step 4 above.

The objects delivered with the MBMS download delivery method shall be formatted according to the announcement in the MPD. The MPD and the described Media Presentation should conform to a profile specified in TS26.247 [32].

Furthermore, the Media Presentation Description fragment may contain reference(s) to Initialisation Segment Description fragment(s) as defined in [98].

Clause 4.4.3 of this specification enables integrity and/or confidentiality protection of MBMS user services data according to 3GPP TS 33.246 [20]. In this case each DASH formatted file is protected using the Protection of Download Data as described in 3GPP TS 33.246 [20].

As this protection mechanism is performed in the underlying layer of the DASH client it is transparent to DASH client and not reflected in the MPD associated to the DASH representation.

For HTTP streaming, QoE reporting on MBMS level can be activated as described in section 8.3.2.1 or 8.3.2.2, and QoE reporting shall in such case be done as specified in section 8.4. The Network Resource, Loss of Objects, and Distribution of Symbol Count Underrun for Failed Blocks QoE metrics are relevant to 3GP-DASH over MBMS.

QoE reporting can also be activated on DASH level as specified in 3GP-DASH [98], section 10 or Annex F, and reporting shall in such case be done according to 3GP-DASH [98].

NOTE: One way of supporting the delivery of a subset of the nominally requested content by the DASH client which indicates explicit willingness to accept such incomplete content, and based on a specific UE implementation architecture, is described in clause 7.2.3 in TR 26.946 [110].

6 Introduction on Delivery Methods

Two delivery methods are defined in the present document - the download delivery method and the streaming delivery method. MBMS delivery methods make use of MBMS bearers for content delivery but may also use the associated procedures defined in clause 9.

Use of MBMS bearers by the download delivery method is described in clause 7. The File Repair Procedure and the Reception Reporting Procedure (described in clause 9) may be used by the download delivery method.

Use of MBMS bearers by the streaming delivery method is described in clause 8.

7 Download Delivery Method

7.1 Introduction

MBMS download delivery method uses the FLUTE protocol (RFC 3926 [9]) when delivering content over MBMS bearers. MBMS download delivery method may use OMA PUSH [79] when delivering content over other UMTS/EPS bearers. Usage of FLUTE protocol is described in clause 7.2. The Usage of OMA Push is described in clasue 7.4. The FLUTE session set-up with RTSP is defined in clause 7.5.

FLUTE is built on top of the Asynchronous Layered Coding (ALC) protocol instantiation (RFC 3450 [10]). ALC combines the Layered Coding Transport (LCT) building block [11], a congestion control building block and the Forward Error Correction (FEC) building block ([12]) 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 [10]), congestion control is not appropriate in the type of environment that MBMS download delivery is provided, and thus congestion control is not used for MBMS download delivery. See figure 10 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.

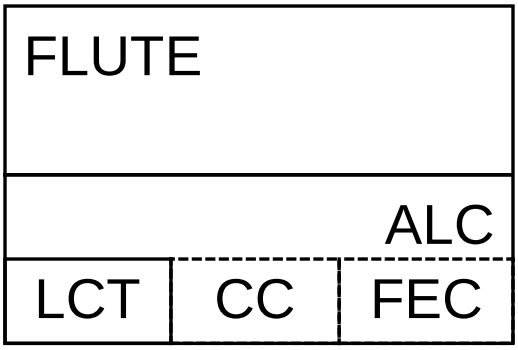


Figure 10: Building block structure of FLUTE

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.

NOTE: One way of supporting the delivery of a subset of the nominally requested content by the DASH client which indicates explicit willingness to accept such incomplete content, and based on a specific UE implementation architecture, is described in clause 7.2.3 in TR 26.946 [110].

7.2 FLUTE usage for MBMS download

The purpose of download is to deliver content in files. In the context of MBMS download, a file contains any type of MBMS data (e.g. 3GPP file (Audio/Video), Binary data, Still images, Text, Service Announcement metadata).

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

UE applications for MBMS user services built upon the download delivery method have three general approaches to getting files from the FLUTE receiver for a joined session:

- **Promiscuous:** Instruct FLUTE to promiscuously receive all files available. Promiscuous reception can be suitable for single purpose sessions (generally with limited number and/or size of files) although uncertainty over the quality and content of files makes this approach generally undesirable.
- **One-copy:** Instruct FLUTE to receive a copy of one or more specific files (identified by the fileURI) - and potentially leaving the session following reception of one copy of all the specified files. Specifying the download file ensures that the UE has an upper bound to the quantity of files downloaded. One-copy reception requires prior knowledge of the file identifiers (fileURIs).
- **Keep-updated:** Instruct FLUTE to receive one or more specific files and continue to receive any updates to those files. As with one-copy, the keep-updated approach bounds the quantity of files downloaded and requires prior knowledge of the file identifiers.

NOTE: The present document does not prevent or endorse changing download reception approach, and any related file list, during the life of the download session. Discovery of session content lists (including file lists) out-of-band of the delivery method sessions is beyond the scope of the present document.

The interaction of these file download modes and the caching directives is defined in section 7.2.13.

MBMS clients and servers supporting MBMS download shall implement the FLUTE specification (RFC 3926 [9]), as well as ALC (RFC 3450 [10]) and LCT (RFC 3451 [11]) 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 MBMS transmission session. It is therefore recommended, that each MBMS transmission session should contain one or more repetitions of the same FDT instance.

7.2.1 Fragmentation of Files

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

7.2.2 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 7.2.12.

A UE that supports MBMS User Services shall support a decoder for the Raptor FEC scheme.

If a UE that supports MBMS User Services receives a mathematically sufficient set of encoding symbols generated according to the encoder specification in [91] for reconstruction of a source block then the decoder shall recover the entire source block. Note that the example decoder described in [91] clause 5.5 fulfils this requirement.

7.2.3 Blocking Algorithm

In the case of the Compact No-Code FEC scheme [12] (FEC Encoding ID 0), then the "Algorithm for Computing Source Block Structure" described within the FLUTE specification (RFC 3926 [9]) shall be used.

In the case of Raptor forward error correction, then the algorithm defined in [91] shall be used.

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

7.2.4 Congestion Control

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

7.2.5 Content Encoding of Files for Transport

Files may be content encoded for transport, as described in [9], in the Download delivery method using the generic GZip algorithm as specified in RFC 1952 [42]. UEs shall support GZip content decoding of FLUTE files (GZIP RFC 1952 [42], clause 9).

7.2.6 Transport File Grouping

Files downloaded as part of a multiple-file delivery are generally related to one another. Examples include web pages, software packages, and the referencing metadata envelopes and their metadata fragments. FLUTE clients analyse the XML-encoded FDT Instances as they are received, identify each requested file, associate it with FLUTE packets (using the TOI) and discover the relevant in-band download configuration parameters of each file.

An additional "group" field in the FLUTE FDT instance and file elements enables logical grouping of related files. A FLUTE receiver should download all the files belonging to all groups where one or more of the files of those groups have been requested. However, a UE may instruct its FLUTE receiver to ignore grouping to deal with special circumstances, such as low storage availability.

The group names are allocated by the FLUTE sender and each specific group name shall group the corresponding files together as one group, including files describes in the same and other FDT Instances, for a session.

Group field usage in FDT Instances is shown in the FDT XML schema (clause 7.2.10). Each file element of an FDT Instance may be labelled with zero, one or more group names. Each FDT Instance element may be labelled with zero, one or more group names which are inherited by all files described in that FDT Instance.

7.2.7 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 “MBMS FEC scheme”, the FEC Payload ID shall be set according to Section 7.2.10 below.

7.2.8 Signalling of Parameters with FLUTE Extension Headers

FLUTE extension header fields EXT_FDT, EXT_FTI , EXT_CENC [9] shall be used 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.

In FLUTE the following applies:

- 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.

7.2.9 Signalling of Parameters with FDT Instances

The extended FLUTE FDT instance schema defined in clause 7.2.10.1 (based on the one in RFC 3926 [9]) 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 file instance).
- Expires (expiry data for 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 file length in bytes).
- Content-Type (content MIME type).
- FEC Encoding ID.

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

NOTE: 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 1: RFC 3926 [9] 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 MBMS:

- Complete (the signalling that an FDT Instance provides a complete, and subsequently unmodifiable, set of file 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 file server should indicate the MD5 hash value whenever multiple versions of the file are anticipated for the download session.

NOTE 2: 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.

7.2.10 FDT Schema

7.2.10.1 Extended FLUTE FDT Schema

The below XML Schema shall be use for the FDT Instance.

This schema extends the schema defined in clause 7.2.10.3 by importing the 3GPP extensions specified in clauses 7.2.10.2, 7.2.10.5, 7.2.14 and 7.2.15. The various schema file names are as follows:

- Schema in clause 7.2.10.1: FLUTE-FDT-3GPP-Main.xsd
- Schema in clause 7.2.10.2: FLUTE-FDT-3GPP-2005-Extensions.xsd
- Schema in clause 7.2.10.5: FLUTE-FDT-3GPP-2007-Extensions.xsd
- Schema in clause 7.2.14: FLUTE-FDT-3GPP-2008-Extensions.xsd
- Schema in clause 7.2.15: FLUTE-FDT-3GPP-2009-Extensions.xsd
- Schema in clause 7.2.10.2: FLUTE-FDT-3GPP-2012-Extensions.xsd
- Schema in clause J.2: schema-version.xsd

In this version of the specification the network shall set the *schemaVersion* element, defined as a child of *FDT-Instance* element, to 2.

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 a UE 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 UE supports one or more versions of the FDT schema with the schema *version* attribute, then the UE 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;

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:IETF:metadata:2005:FLUTE:FDT"
    xmlns:fl="urn:IETF:metadata:2005:FLUTE:FDT"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mbms2005="urn:3GPP:metadata:2005:MBMS:FLUTE:FDT"
    xmlns:mbms2007="urn:3GPP:metadata:2007:MBMS:FLUTE:FDT"
    xmlns:mbms2008="urn:3GPP:metadata:2008:MBMS:FLUTE:FDT_ext"
    xmlns:mbms2009="urn:3GPP:metadata:2009:MBMS:FLUTE:FDT_ext"
    xmlns:mbms2012="urn:3GPP:metadata:2012:MBMS:FLUTE:FDT"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    targetNamespace="urn:IETF:metadata:2005:FLUTE:FDT"
    elementFormDefault="qualified"
    version="2">
    <xs:import namespace="urn:3GPP:metadata:2005:MBMS:FLUTE:FDT"
                schemaLocation="FLUTE-FDT-3GPP-2005-Extensions.xsd"/>
    <xs:import namespace="urn:3GPP:metadata:2007:MBMS:FLUTE:FDT"
                schemaLocation="FLUTE-FDT-3GPP-2007-Extensions.xsd"/>
    <xs:import namespace="urn:3GPP:metadata:2008:MBMS:FLUTE:FDT_ext"
                schemaLocation="FLUTE-FDT-3GPP-2008-Extensions.xsd"/>
    <xs:import namespace="urn:3GPP:metadata:2009:MBMS:FLUTE:FDT_ext"
                schemaLocation="FLUTE-FDT-3GPP-2009-Extensions.xsd"/>
    <xs:import namespace="urn:3GPP:metadata:2012:MBMS:FLUTE:FDT"
                schemaLocation="FLUTE-FDT-3GPP-2012-Extensions.xsd"/>
    <xs:import namespace="urn:3gpp:metadata:2009:MBMS:schemaVersion"
                schemaLocation="schema-version.xsd"/>
```

```
    <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="sv:schemaVersion"/>
            <xs:element ref="mbms2012:Base-URL-1" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="mbms2012:Base-URL-2" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="sv:delimiter"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Group" type="mbms2005:groupIdType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="MBMS-Session-Identity-Expiry" type="mbms2005:MBMS-Session-Identity-
Expiry-Type" 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:attribute ref="mbms2008:FullFDT" use="optional" default="false"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="FileType">
        <xs:sequence>
            <xs:element ref="mbms2007:Cache-Control" minOccurs="0"/>
            <xs:element ref="sv:delimiter"/>
            <xs:element ref="mbms2012:Alternate-Content-Location-1" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element ref="mbms2012:Alternate-Content-Location-2" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element ref="sv:delimiter"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="Group" type="mbms2005:groupIdType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="MBMS-Session-Identity" type="mbms2005:MBMS-Session-Identity-Type"
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 ref="mbms2009:Decryption-KEY-URI" use="optional"/>
        <xs:attribute ref="mbms2012:FEC-Redundancy-Level" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
</xs:schema>
```

7.2.10.2 3GPP FDT Extension Type Schema

The extension of the IETF FLUTE FDT schema is done using the following schema definition:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3GPP:metadata:2005:MBMS:FLUTE:FDT"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3GPP:metadata:2005:MBMS:FLUTE:FDT"
    elementFormDefault="qualified">
    <xs:complexType name="MBMS-Session-Identity-Expiry-Type">
```

```
        <xs:simpleContent>
            <xs:extension base="MBMS-Session-Identity-Type">
                <xs:attribute name="value" type="xs:unsignedInt" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="MBMS-Session-Identity-Type">
        <xs:restriction base="xs:unsignedByte"/>
    </xs:simpleType>
    <xs:simpleType name="groupIdType">
        <xs:restriction base="xs:string"></xs:restriction>
    </xs:simpleType>
</xs:schema>
```

The Release 11 extension of the FLUTE FDT schema is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3GPP:metadata:2012:MBMS:FLUTE:FDT"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:ns1="urn:3GPP:metadata:2012:MBMS:FLUTE:FDT"
    targetNamespace="urn:3GPP:metadata:2012:MBMS:FLUTE:FDT"
    elementFormDefault="qualified">
    <xs:element name="Alternate-Content-Location-1" type="Alternative-Content-LocationType"/>
    <xs:element name="Alternate-Content-Location-2" type="Alternative-Content-LocationType"/>
    <xs:complexType name="Alternative-Content-LocationType">
        <xs:sequence>
            <xs:element name="Alternate-Content-Location" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="Availability-Time" type="xs:dateTime"/>
    </xs:complexType>
    <xs:element name="Base-URL-1" type="xs:anyURI"/>
    <xs:element name="Base-URL-2" type="xs:anyURI"/>
    <xs:attribute name="FEC-Redundancy-Level" type="xs:unsignedInt"/>
</xs:schema>
```

7.2.10.3 IETF FDT Schema

Below is the IETF based FDT XML schema that has been extended to define the main FDT schema in sub-clause 7.2.10.1.

NOTE 1: As the schema in RFC 3926 is not valid there exist no stable reference, thus this specification will include this schema until IETF has published an updated version of the schema.

NOTE 2: The schema in this sub-clause is provided for information, since the extended schema of sub-clause 7.2.10.1 is copying all the schema of this sub-clause and adds 3GPP specific extensions to it.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:IETF:metadata:2005:FLUTE:FDT"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:IETF:metadata:2005:FLUTE:FDT"
    elementFormDefault="qualified">
    <xs:element name="FDT-Instance" type="FDT-InstanceType"/>
    <xs:complexType name="FDT-InstanceType">
        <xs:sequence>
            <xs:element name="File" type="FileType" maxOccurs="unbounded"/>
            <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: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:anyAttribute processContents="skip"/>
    </xs:complexType>
</xs:schema>
```

7.2.10.4 Example of FDT

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance
    xmlns="urn:IETF:metadata:2005:FLUTE:FDT"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:mbms2007="urn:3GPP:metadata:2007:MBMS:FLUTE:FDT"
    xmlns:mbms2008="urn:3GPP:metadata:2008:MBMS:FLUTE:FDT_ext"
    xmlns:mbms2009="urn:3GPP:metadata:2009:MBMS:FLUTE:FDT_ext"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:IETF:metadata:2005:FLUTE:FDT FLUTE-FDT-3GPP-Main.xsd"
    FEC-OTI-FEC-Encoding-ID="1"
    Complete="true"
    Content-Encoding="gzip"
    FEC-OTI-Encoding-Symbol-Length="512"
    Expires="331129600"
    mbms2008:FullFDT="true">
    <File
        Content-Type="application/sdp"
        Content-Length="7543"
        Transfer-Length="4294"
        TOI="2"
        FEC-OTI-Encoding-Symbol-Length="16"
        FEC-OTI-Scheme-Specific-Info="AAEBBA=="
        Content-Location="http://www.example.com/fancy-session/main.sdp"
        mbms2009:Decryption-KEY-URI="http://www.example.com/key-uri">
        <mbms2007:Cache-Control>
            <mbms2007:Expires>30</mbms2007:Expires>
        </mbms2007:Cache-Control>
        <sv:delimiter>0</sv:delimiter>
        <sv:delimiter>0</sv:delimiter>      <MBMS-Session-Identity>93</MBMS-Session-Identity>
    </File>
    <File
        Content-Type="String"
        Content-Length="161934"
        Transfer-Length="157821"
        TOI="3"
        FEC-OTI-Encoding-Symbol-Length="512"
        Content-Location="http://www.example.com/fancy-session/trailer.3gp">
        <sv:delimiter>0</sv:delimiter>
        <sv:delimiter>0</sv:delimiter>
        <MBMS-Session-Identity>93</MBMS-Session-Identity>
    </File>
    <sv:schemaVersion>2</sv:schemaVersion>
    <sv:delimiter>0</sv:delimiter>
    <MBMS-Session-Identity-Expiry value="3311288760">93</MBMS-Session-Identity-Expiry>
</FDT-Instance>
```

7.2.10.5 3GPP FDT Extensions

The following schema defines the new elements

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3GPP:metadata:2007:MBMS:FLUTE:FDT"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3GPP:metadata:2007:MBMS:FLUTE:FDT"
    elementFormDefault="qualified">

    <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>
</xs:schema>
```

7.2.10.6 FEC Redundancy Level Extension

A MBMS download service may indicate FEC redundancy level in FLUTE FDT instance. The attribute " FEC-Redundancy-Level" is included within element "file" of the FDT to indicate the FEC redundancy level for the file. For example,if the FEC-Redundancy-Level is set to 20, it means BM-SC will add 20% extra redunancy for this file during MBMS delivery.

The FEC redundancy level information may be used by the MBMS client to e.g. stop decoding a particular DASH segment when MBMS client detects that the packet loss rate already exceeds the FEC redundancy level for that segment.

The XML syntax of the " FEC-Redundancy-Level" attribute within the FLUTE FDT is specified in clause 7.2.10.2, and the attribute is further refered to by the main FDT schema of clause 7.2.10.1.

7.2.11 MBMS Session Identity

The *MBMS-Session-Identity* element associates the file to the identity of the MBMS session. If the file will be part of several MBMS transmission sessions, then a list of MBMS session identities is defined.

The *MBMS-Session-Identity-Expiry* element associates an expiration time with a MBMS session identity value. Similar to the FLUTE FDT expiration time, the MBMS session identity expiration time (*value* attribute) is expressed within the FDT Instance payload as a 32 bit data field. The value of the data field represents the 32 most significant bits of a 64 bit Network Time Protocol (NTP) [78] time value. These 32 bits provide an unsigned integer representing the time in seconds relative to 0 hours 1 January 1900.

7.2.12 FEC Scheme definition

7.2.12.1 General

This clause defines an FEC encoding scheme for the MBMS forward error correction code defined in [91] for the download delivery 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 [91], sub-clauses 3.1 and 3.2 respectively.

7.2.13 Caching Directives

A file download service may indicate the caching recommendations for a specific file or set of files that are delivered using FLUTE. The caching directives are to be used with the file download modes as follows:

- Promiscuous mode: it is recommended to use the caching directives with the promiscuous mode as it enables improved management of the storage at the UE. Applications make use of available copies of files as long as their respective caching time is still valid. In case one or several files have expired and the download session is

still available, the UE should join the FLUTE session and download the expired files. Alternatively, the UE may attempt to retrieve the file using HTTP and the file URL.

- One-Copy mode: Caching directives may be used with the one-copy mode to indicate the validity of a certain file. Applications requesting the file will receive the cached file as long as it is still valid. A file that is not expected to be static may indicate a long expiry time or permanent validity.

- Keep-Updated mode: it is recommended to use the caching directives with the keep-updated mode to indicate the validity of a certain file. Applications requesting the file will receive the cached file as long as it is still valid.

The caching functionality defines three different caching directives:

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

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

- **Expires**: this directive is used by the server to indicate the expected expiry time of a specific file (or set of files). It indicates a date and time value in the HTTP date format or in the NTP timestamp format.

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

7.2.14 Indicating a full FDT snapshot

If the server wants to inform the client about the current FDT snapshot, the server shall set the “FullFDT” attribute in the FLUTE FDT instance file. If the “FullFDT” attribute is set, the FDT instance shall be equivalent to the full File Delivery Table. Note FDT instances with a higher FDT instance ID may again extend the File Delivery Table.

A new attribute "FullFDT" is created within the element "FDT-Instance" of the FDT to indicate to the receivers that the FDT Instance contains the exact set of Transport Objects that are currently scheduled for transmission by the sender, in the actual FLUTE session.

The XML syntax of the "FullFDT" attribute within the FLUTE FDT is the following.

```
  <?xml version="1.0" encoding="UTF-8"?>
  <xs:schema xmlns="urn:3GPP:metadata:2008:MBMS:FLUTE:FDT_ext"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   targetNamespace="urn:3GPP:metadata:2008:MBMS:FLUTE:FDT_ext"
   elementFormDefault="qualified">

   <xs:attribute name="FullFDT" type="xs:boolean"
 />
  </xs:schema>
```

This attribute differs from the existing "Complete" attribute in that the "Complete" attribute indicates that no new objects description will be provided in future FDT Instances within this session.

No assumption shall be made about the fact that a given FDT instance for which the attribute "FullFDT" is absent or set to FALSE, contains the exact set of Transport Objects that are currently scheduled for transmission by the sender, in the actual FLUTE session.

When two FDT instances with attribute "FullFDT" is equal to TRUE are received by a receiver and valid in a given time (that is to say they have not expired), the FDT instance with the highest FDT Instance ID shall be used by the terminal.

7.2.15 Decryption key indicating of protected download data

A MBMS download service may indicate relevant decryption key file for protected download file in FLUTE FDT instance. A new attribute "Decryption-KEY-URI" is created within element "file" of the FDT to indicate the association between protected download file and relevant decryption key file. The value of "Decryption-KEY-URI" in "file" element shall be equal to the content-location of the MIKEY file that contains the decryption key file.

When the server delivers a protected download file, the server should set a "Decryption-KEY-URI" field in the corresponding file element in the FLUTE FDT instance. When a UE receives a protected file, the UE may instruct its

FLUTE receiver to download the relevant decryption key file according to "Decryption-KEY-URI" field in file element of FDT instance.

The XML syntax of the "Decryption-KEY-URI" attribute within the FLUTE FDT is the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2009:MBMS:FLUTE:FDT_ext"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 targetNamespace="urn:3GPP:metadata:2009:MBMS:FLUTE:FDT_ext"
 elementFormDefault="qualified">
 <xs:attribute name="Decryption-KEY-URI" type="xs:anyURI"/>
</xs:schema>
```

7.3 SDP for Download Delivery Method

7.3.1 Introduction

RFC 3926 [9] describes required and optional parameters for FLUTE session and media descriptors. This clause specifies SDP for FLUTE session that is used for the MBMS download and service announcement sessions. The formal specification of the parameters is given in ABNF ([23]).

7.3.2 SDP Parameters for MBMS download session

The semantics of a Session Description of an MBMS download 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.
- Mode of MBMS bearer per media.
- FEC capabilities and related parameters.
- Service-language(s) per media.

This list includes the parameters required by FLUTE - RFC 3926 [9]

These shall be expressed in SDP ([14] and [15]) syntax according to the following clauses.

7.3.2.1 Sender IP address

There shall be exactly one IP sender address per MBMS download session, and thus there shall be exactly one IP source address per complete MBMS download session SDP description. The IP source address shall be defined according to the source-filter attribute ("a=source-filter:") ([14] and [15]) 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 MBMS download 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 MBMS download session employs only a single LCT (multicast) channel.

7.3.2.2 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.

7.3.2.3 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 ([14]). The destination port number shall be defined according to the <port> sub-field of the media announcement field ("m=") of SDP ([14]).

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.

7.3.2.4 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 [11]).

The TSI shall be defined according 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
```

7.3.2.5 Multiple objects transport indication

RFC 3626 [9] 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.

7.3.2.6 Session Timing Parameters

A MBMS download session start and end times shall be defined according to the SDP timing field ("t=") ([14]).

7.3.2.7 Mode of MBMS bearer per media

A new MBMS bearer mode declaration attribute is defined which results in, e.g.:

- a=mbms-mode:broadcast 123869108302929 1

OR

- a=mbms-mode:broadcast-sfn 123869108302929

The MBMS bearer mode declaration attribute shall be used in session descriptions using one or more MBMS broadcast mode media or broadcast-sfn mode media. If all media declarations use MBMS broadcast mode or broadcast-sfn mode, 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 occurance of the "mbms-mode" attribute. If one or more media using MBMS multicast mode is present in the same declaration as media using MBMS broadcast mode, then only media using the MBMS broadcast mode or broadcast-sfn mode will contain the "mbms-mode" attribute.

mbms-bearer-mode-declaration-line = "a=mbms-mode:" ("broadcast" SP tmgi SP mbms-counting-information) / (”broadcast-mbsfn” SP tmgi) CRLF

tmgi = 1*15DIGIT

mbms-counting-indication = 1 * DIGIT

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)
MBMS 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 [40] including the coding of the fields. Octets 3 to 8 (MBMS Service ID, MCC and MNC) shall be placed in the tmgi attribute of the MBMS bearer mode 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 MBMS Service ID field may be omitted.

The MBMS Counting Information (mbms-counting-information) information element is defined in TS 25.413 [87] and indicates whether the RAN level counting procedures are applicable or not for the MBMS broadcast mode. The value 0 corresponds to the information element value of “not counting” and the value 1 corresponds to the information element value “counting”.

7.3.2.8 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 download delivery 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 UE by other means, the UE 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

7.3.2.9 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 [74] and ISO 3166 [75] (e.g. "a=lang:EN-US"). These are the same tags used in the User Service Description XML.

7.3.2.10 Bandwidth Specification

The maximum bit-rate required by this FLUTE session shall be specified using the "AS" bandwidth modifier [14] 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.

7.3.3 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=File delivery session example
i=More information
t=2873397496 2873404696
a=mbms-mode:broadcast 123869108302929 1
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=64
a=lang:EN
a=FEC:0
```

7.4 OMA Push usage for MBMS Download

7.4.1 Introduction

OMA Push may be used for MBMS download reception when MBMS Bearers are not available. The MBMS UE registers its MSISDN with the BM-SC to receive the Download Sessions using OMA Push. The BM-SC distributes FLUTE FDT instance which allows the MBMS UE to fetch files of interest.

If the MBMS UE is out of its home network and if at least one *unicastAccessURI* element is available in the deliver method description, the MBMS UE should register its MBMS Download Services with the BM-SC.

7.4.2 HTTP registration and deregistration procedure

The MBMS UE may register and deregister for unicast service delivery, if the MBMS User Service Description for this service includes at least one *unicastAccessURI* element in the *deliveryMethod* element.

The HTTP (RFC 2616 [18]) GET method is used for this purpose. If more than one *unicastAccessURI* is provided in the *deliveryMethod* element, the UE shall randomly select one.

In the following, we give the details of the syntax used for the above request method in ABNF [23].

- unicast_access_request_http_URL = unicast_access_URI "?" query
- unicast_access_URI = <*unicastAccessURI* from the User Service Description; URI-reference is as defined in [19].>
- query = action “&” serviceId "&" msisdn
- action = “action=” (“register” | “Register” | “deregister” | “Deregister”)
- serviceId = "serviceId=" <value of the *serviceId* attribute of the User Service Description>
- msisdn = “msisdn=” 1*DIGIT <format as defined in [77]

The BM-SC responds with an “200 OK” status code in case of successful registration or deregistration. With the response to a successful registration request, an associated delivery procedure fragment as defined in clause 9.5 shall be delivered to the MBMS UE. The MBMS UE uses the File Repair procedure as described in the received associated delivery procedure description fragment. The file repair procedure is defined in clause 9.3. Note, the file repair procedure allows also to fetch complete files.

An HTTP GET request with “action” value set to “register” or “Register” shall be sent to register the MBMS UE for unicast file delivery service. The request shall also include the *serviceId* and the MSISDN of the MBMS UE. The format for the MSISDN is defined in [77]. The following is an exmaple for a registration request:

GET /unicasDelivery?action=Register&serviceId=urn:3gpp:0010120123hotdog&MSISDN=436642012345 **HTTP/1.1**

Host: bmsc.example.com

An HTTP GET request with “action” value set to “deregister” or “Deregister” shall be sent to deregister the MBMS UE from the unicast file delivery service. The request also includes the service ID and the MSISDN number of the MBMS UE as shown in the following example:

GET /unicasDelivery?action=Deregister&serviceId=urn:3gpp:0010120123hotdog&MSISDN=436642012345 **HTTP/1.1**

Host: bmsc.example.com

7.4.3 MBMS Download Delivery Method over OMA push bearers

MBMS Download over OMA Push bearers are formatted according to the OMA Push OTA specification [79].

OTA-WSP shall be used over unicast bearers. Application port addressing shall be used as specified in [79]. The application ID to be used is 0x9045 as allocated by OMNA [85].

OTA-HTTP may be used over the HTTP push bearer. Application port addressing shall be used as specified in [79]. The application ID to be used is 0x9045 as allocated by OMNA [85].

The Content-Encoding header shall be included if GZip is used.

The MBMS UE receives the FLUTE FDT instance and the Download Header instance via OMA Push OTA protocol [79]. Both documents are encapsulated in a multipart MIME document. Optionally an associated delivery description fragment as defined in clause 9.5 is part of the pushed document. The FLUTE FDT instance is identified by the MIME type “application/fdt+xml” and the associated delivery description fragment by the mimetype as defined in Annex C.7. The download header instance should use a default mime type “application/xml”.

The XML schema for the download header fragment is defined below. The *serviceId* element contains the unique identifier of the MBMS service. The MBMS UE uses the Service Id to select the target application and has received the *serviceId* with the User Service Annoucement fragment. The format of the *serviceId* element is defined in clasue 11.2.1.1.

The *fdtInstanceId* element shall contain the FDT instance Identifier for the sent FDT instance. Note, the FDT instance id is transferred using the FDT Instance Header as defined in clause 3.4.1 of [9]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns="urn:3GPP:metadata:2007:MBMS:downloadHeader"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3GPP:metadata:2007:MBMS:downloadHeader"
elementFormDefault="qualified"
attributeFormDefault="unqualified">

<xs:element name="mbmsDownloadHeader">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="serviceId" type="xs:anyURI"/>
                <xs:element name="fdtInstanceId" type="xs:unsignedInt"/>
                <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

The UE will then have necessary information about all files in the FLUTE session including their fileURIs, content encodings, content lengths etc.

7.5 FLUTE session setup and control with RTSP

7.5.1 Introduction

In case the MBMS User Service contains MBMS Streaming and MBMS Download delivery sessions, it may be beneficial to control all flows with RTSP. The prime use case of FLUTE session set-up and control with RTSP is for sending MBMS streaming associated presentation data.

7.5.2 SDP handling

The FLUTE specific SDP extensions are defined in clause 7. For the FLUTE session establishment using RTSP, a control URI as defined in [88] shall be present for the FLUTE media description. Note, a control URI is defined by the "a=control:" SDP field according to [88].

7.5.3 RTSP SETUP Method

The control URI as defined in [88] shall be present for each FLUTE media description in the SDP. The control URI is used within the RTSP SETUP method to establish the described FLUTE sessions.

The RTSP transport protocol specifier for FLUTE as defined in [88] shall be "FLUTE/UDP". One and only one UDP port is allocated for each FLUTE channel.

The following RTP specific parameters shall be used in the transport request and responds header for FLUTE sessions:

- client_port: This parameter provides the unicast FLUTE port(s) on which the client has chosen to receive FLUTE data.
- server_port: This parameter provides the unicast FLUTE port(s) on which the server has chosen to send data.

7.5.4 RTSP PLAY Method

The PLAY method tells the server to start sending data including FLUTE session data as defined in [88]. The RTSP server forwards the FLUTE packets as according by the RTSP range header in the RTSP PLAY.

Only ntp and clock range units may be used with the "Range" headers. Normal Play Time (NPT) indicates the stream absolute position relative to the beginning of the presentation. The NPT consists of a decimal fraction.The clock range header describe the absolute time expressed as ISO 8601 timestamps, using UTC (GMT).

7.5.5 RTSP PAUSE Method

The PAUSE request causes the stream delivery including all FLUTE sessions to be interrupted (halted) as defined in [88].

7.5.6 RTSP TEARDOWN Method

The TEARDOWN client to server request stops the stream delivery including all FLUTE data delivery for the given URI, freeing the resources associated with it. Details for the TEARDOWN method are defined in [88].

8 Streaming delivery method

8.1 Introduction

The purpose of the MBMS streaming delivery method is to deliver continuous multimedia data (i.e. speech, audio, video and DIMS) over an MBMS bearer. Using MBMS Streaming delivery on unicast is defined in clause 8.5. This delivery method complements the download delivery method which consists of the delivery of files. The streaming delivery method is particularly useful for multicast and broadcast of scheduled streaming content.

8.2 Transport protocol

RTP is the transport protocol for MBMS streaming delivery. RTP provides means for sending real-time or streaming data over UDP and is already used for the transport of PSS in 3GPP. RTP provides RTCP for feedback about the transmission quality. The transmission of RTCP packets in the downlink (sender reports) is allowed. In this version of the specification, RTCP RR shall be turned off by SDP RR bandwidth modifiers. Note that in the context of MBMS detection of link aliveness is not necessary.

8.2.1 RTP payload formats for media

The RTP payload formats and corresponding MIME types are closely aligned with those defined in PSS [47] . For RTP/UDP/IP transport of continuous media the following RTP payload formats shall be used:

- AMR narrow-band speech codec (see sub-clause 10.2) RTP payload format according to RFC 4867 [33]. A MBMS client is not required to support multi-channel sessions.
- AMR wideband speech codec (see sub-clause 10.2) RTP payload format according to RFC 4867 [33]. A MBMS client is not required to support multi-channel sessions.
- Extended AMR-WB codec (see sub-clause 10.3) RTP payload format according to [34].
- Enhanced aacPlus codec (see sub-clause 10.3) RTP payload format and MIME types according to RFC 3640 [41], namely the Low Bit-Rate AAC or the High Bit-Rate AAC modes.

- H.264 (AVC) video codec (see sub-clause 10.5) RTP payload format according to [35]. An MBMS client supporting H.264 (AVC) is required to support all three packetization modes: single NAL unit mode, non-interleaved mode and interleaved mode. For the interleaved packetization mode, an MBMS client shall support streams for which the value of the "sprop-deint-buf-req" MIME parameter is less than or equal to MaxCPB * 1000 / 8, inclusive, in which "MaxCPB" is the value for Video Coding Layer (VCL) parameters of the H.264 (AVC) profile and level in use, as specified in [43].
- DIMS (see sub-clause 10.12) RTP payload format according to [89].
- Timed Text (see sub-clause 10.10) RTP payload format according to [93].

8.2.2 FEC mechanism for RTP

8.2.2.0 General

The “MBMS FEC scheme” is the fully-specified FEC scheme defined in [106], section 6 with ID 1.

The source flows for the MBMS FEC scheme are UDP flows including RTP, RTCP, SRTP and MIKEY packets. The payload of such UDP packets constitute an Application Data Unit (ADU) as defined in RFC6363 [107]. The source data flow with which the ADUs are associated is the UDP flow identity of the corresponding UDP flow.

A UE that supports MBMS User Services shall support a decoder for the “MBMS FEC scheme”. The use of MBMS FEC by the sender is recommended, but it is permitted not to use it. In the case where the FEC is not used by the sender, the FEC Layer should not be used (i.e. RTP is mapped onto UDP directly).

The mechanism does not place any restrictions on the source data which can be protected together, except that the source data is carried over UDP. The data may be from several different UDP flows that are protected jointly.

A UE supporting the streaming delivery method shall support the packet format for FEC packets..

If any FEC source packets have been lost, but sufficient FEC source and FEC repair packets have been received, FEC decoding can be performed to recover the FEC source block. The original packets UDP payload and UDP flow identity can then be extracted from the source block and provided to the upper layer. If not enough FEC source and repair packets were received, only the original packets that were received as FEC source packets will be available. The rest of the original packets are lost.

If a UE that supports MBMS User Services receives a mathematically sufficient set of encoding symbols generated according to the encoder specification in RFC5053 [91], section 5.3, for reconstruction of a source block, then the decoder shall recover the entire source block. Note that the example decoder described in [91] clause 5.5 fulfils this requirement.

Note that the receiver must be able to buffer all the original packets and allow time for the FEC repair packets to arrive and FEC decoding to be performed before media playout begins. The min-buffer-time parameter specified in sub-clause 8.3.1.8 helps the receiver to determine a sufficient duration for initial start-up delay.

The protocol architecture is illustrated in figure 11.

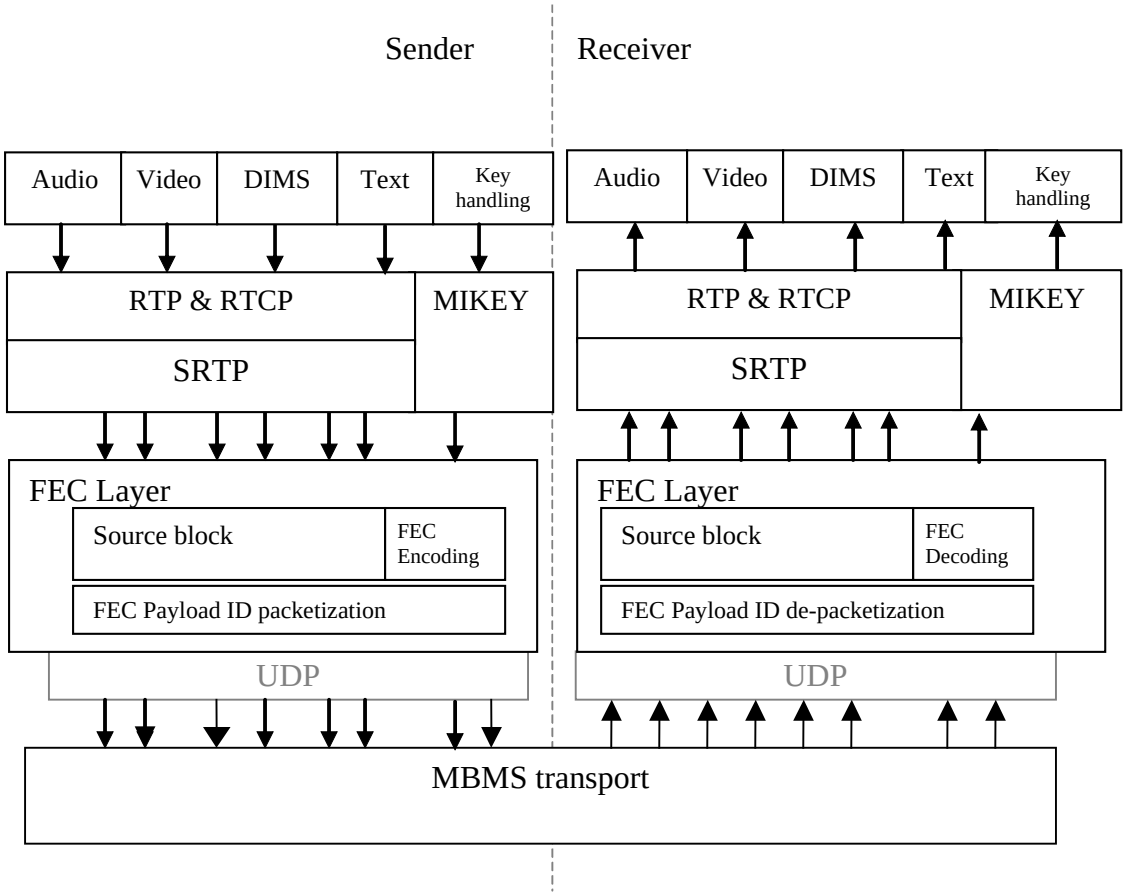


Figure 11: FEC mechanism for the streaming delivery method interaction diagram

Figure 11 depicts how one or more out of several possible packet flows of different types (Audio, video, DIMS, text RTP and RTCP flows, MIKEY flow) are sent to the FEC layer for protection. The source packets are modified to carry the FEC payload ID and a new flow with repair data is generated. The receiver takes the source and repair packets and buffers them to perform, if necessary, the FEC decoding. After appropriate buffering received and recovered source packets are forwarded to the higher layers. The arrows in the figure indicate distinct data flows.

8.2.2.1 Sending Terminal Operation (Informative)

It is assumed that the sender has constructed or received original data packets for the session. These may be RTP, RTCP, MIKEY or other UDP packets. The following procedures are based on the UDP payload and the identity of the UDP flow. The UDP payload constitutes and ADU according to RFC6363 [107] and the identity of the UDP flow is the integer identifier associated with the identifier of the ADU flow.

In order to FEC protect a sequence of original data packets, the sender constructs a source block as specified in RFC6681 [106], section 5 to which the FEC algorithm is to be applied, and includes the original source packet data within FEC source packets. The following operations describe a possible way to generate compliant FEC source packet and FEC repair packet streams:

1. Each original packet is placed in the source block. In doing so, the Source FEC Payload ID information to be included in the FEC payload ID of the FEC source packet can be determined. In the source block the identity of the packet's flow is marked using the Flow ID. See RFC6681 [106], section 5 for details.

2. The FEC source packet is constructed according to sub-clause 8.2.2.4. The identity of the original flow is maintained by the source packet through the use of the destination UDP port number and destination IP address, which has been advertised (for example using SDP), as carrying FEC source packets generated from an original stream of a particular protocol (e.g. RTP, RTCP, SRTP, MIKEY etc.). See sub-clause 8.2.2.13.

3. The generated FEC source packet is sent using UDP.

When a source block is complete, the FEC encoder generates encoding symbols and places these symbols into FEC repair packets, to be conveyed to the receivers. These repair packets are sent using normal UDP procedures to a unique destination port to separate it from any of the source packet flows.

In particular cases it may be advantageous not to use FEC for some source blocks and to signal this to the receiver. In this case the sender may send one or more empty repair packets consisting exclusively of the Repair FEC Payload ID. This will be helpful in particular for selective FEC where some of the source blocks (e.g. consisting of reference video frames) are FEC protected while others (e.g. consisting exclusively of non-reference frames) will not be protected.

8.2.2.2 Receiving Terminal Operation (Informative)

The following describes a possible receiver algorithm, when receiving an FEC source or repair packet:

1. If a FEC source packet is received (as indicated by the UDP port on which it was received):
   a. The original source packet is reconstructed by removing the Source FEC Payload ID. The resulting packet is buffered to allow time for the FEC repair.
   b. The resulting packet is placed into the source block according to the information in the Source FEC Payload ID and the source block format described in [106], section 5. The UDP port the packet was received on is used to determine the Flow ID written into the source block.
2. If an FEC repair packet is received (as indicated by the UDP port), the contained encoding symbols are placed into an FEC encoding block according to the Repair FEC Payload ID. In case the received FEC repair packet is empty, there are no repair symbols to be placed in the FEC encoding block.
3. If at least one source packet is missing, then FEC decoding may be desirable. The FEC decoder determines if the encoding block constructed in steps 1 and 2 contains enough symbols from the source and repair packets for decoding and, if so, performs the decoding operation. If only empty FEC repair packets are received, the receiver may start immediately some procedures to conceal the effect of missing media data.
4. Any missing source packets that were reconstructed during the decoding operation are then buffered as normal received packets (see step 1a above).

Note that the above procedure may result in that not all original packets are recovered, and they must simply be marked as being lost.

Obviously, buffering and packet re-ordering are required to insert any reconstructed packets in the appropriate place in the packet sequence if that is necessary according to the used higher layer protocol (RTP, RTCP or MIKEY). To allow receivers to determine the minimal start-up buffering requirement for FEC decoding, the min-buffer-time parameter indicates a minimum initial buffering time that is sufficient regardless of the position of the stream in which the reception starts.

8.2.2.3 (Void)

8.2.2.4 Packet format for FEC source packets

The packet format for FEC source packets as defined in RFC6363 [107], section 5.3, shall be used to encapsulate an original UDP packet.

The destination IP address and UDP port shall be set as indicated in the session control signalling. This ensures that the receiver can determine which protocols and FEC Payload ID formats are used for this flow. The remaining fields in the IP and UDP headers shall be set according to their specifications.

The Source FEC Payload ID shall be constructed according to RFC 6681 [106], section 6.2.2.

The FEC Source packets over IP and UDP are indicated to be used for a flow by using one of the SDP protocol identifiers “UDP/MBMS-FEC/RTP/AVP”, “UDP/MBMS-FEC/RTP/SAVP” depending on the upper layer protocol RTP/AVP or RTP/SAVP respectively. If MIKEY is FEC protected and encapsulated in source packets, then it is indicated in the security description using the *fecProtection* element and the destination IP address.

8.2.2.5 Packet Format for Repair packets

The packet format for FEC repair packets as defined in RFC6363 [107], section 5.4 shall be used for repair packets.

The UDP payload consists of the Repair FEC Payload ID, and zero, one or more repair symbols. The format of the Repair FEC payload ID is defined in clause [106], section 6.2.3.

The repair packet sent over IP and UDP is indicated in the SDP using the protocol identifier “UDP/MBMS-REPAIR”.

8.2.2.6 Void

8.2.2.7 FEC block Construction algorithm and example (informative)

This section provides an example how to use the methods in RFC6363 [107] and RFC6681 [106] to generate a source block.

When the original UDP packet is placed into the source block, the value of the UDP flow identifier, F, followed by the value of the UDP payload length, L, are first written as a single byte and two-byte value in network byte order (i.e. with high order byte first) respectively into the first available bytes in the source block, followed by the UDP packet payload itself (i.e. not including the IP/UDP headers). Following this, if the next available byte is not the first byte of a new symbol, then padding bytes up to the next symbol boundary shall be included using the value 0 in each byte. As long as any source UDP packets remain to be placed, the procedure is repeated starting each UDP flow identifier at the start of the next encoding symbol.

An example of forming a source block is given in figure 14 below. In this example, three UDP packets of lengths 26, 52 and 103 bytes have been placed into a source block with symbol size T = 16 bytes. The first two packets are from UDP flow 0 and the third from UDP flow 1. Each entry in Figure 14 is a byte and the rows correspond to the source symbols and are numbered from 0 to 12. $B_{i,j}$ denotes the (j+1)th byte of the (i+1)th UDP packet.

0	26		$B_{0,0}$	$B_{0,1}$	$B_{0,2}$	$B_{0,3}$	$B_{0,4}$	$B_{0,5}$	$B_{0,6}$	$B_{0,7}$	$B_{0,8}$	$B_{0,9}$	$B_{0,10}$	$B_{0,11}$	$B_{0,12}$
$B_{0,13}$	$B_{0,14}$	$B_{0,15}$	$B_{0,16}$	$B_{0,17}$	$B_{0,18}$	$B_{0,19}$	$B_{0,20}$	$B_{0,21}$	$B_{0,22}$	$B_{0,23}$	$B_{0,24}$	$B_{0,25}$	0	0	0
0	52		$B_{1,0}$	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	$B_{1,4}$	$B_{1,5}$	$B_{1,6}$	$B_{1,7}$	$B_{1,8}$	$B_{1,9}$	$B_{1,10}$	$B_{1,11}$	$B_{1,12}$
$B_{1,13}$	$B_{1,14}$	$B_{1,15}$	$B_{1,16}$	$B_{1,17}$	$B_{1,18}$	$B_{1,19}$	$B_{1,20}$	$B_{1,21}$	$B_{1,22}$	$B_{1,23}$	$B_{1,24}$	$B_{1,25}$	$B_{1,26}$	$B_{1,27}$	$B_{1,28}$
$B_{1,29}$	$B_{1,30}$	$B_{1,31}$	$B_{1,32}$	$B_{1,33}$	$B_{1,34}$	$B_{1,35}$	$B_{1,36}$	$B_{1,37}$	$B_{1,38}$	$B_{1,39}$	$B_{1,40}$	$B_{1,41}$	$B_{1,42}$	$B_{1,43}$	$B_{1,44}$
$B_{1,45}$	$B_{1,46}$	$B_{1,47}$	$B_{1,48}$	$B_{1,49}$	$B_{1,50}$	$B_{1,51}$	0	0	0	0	0	0	0	0	0
1	103		$B_{2,0}$	$B_{2,1}$	$B_{2,2}$	$B_{2,3}$	$B_{2,4}$	$B_{2,5}$	$B_{2,6}$	$B_{2,7}$	$B_{2,8}$	$B_{2,9}$	$B_{2,10}$	$B_{2,11}$	$B_{2,12}$
$B_{2,13}$	$B_{2,14}$	$B_{2,15}$	$B_{2,16}$	$B_{2,17}$	$B_{2,18}$	$B_{2,19}$	$B_{2,20}$	$B_{2,21}$	$B_{2,22}$	$B_{2,23}$	$B_{2,24}$	$B_{2,25}$	$B_{2,26}$	$B_{2,27}$	$B_{2,28}$
$B_{2,29}$	$B_{2,30}$	$B_{2,31}$	$B_{2,32}$	$B_{2,33}$	$B_{2,34}$	$B_{2,35}$	$B_{2,36}$	$B_{2,37}$	$B_{2,38}$	$B_{2,39}$	$B_{2,40}$	$B_{2,41}$	$B_{2,42}$	$B_{2,43}$	$B_{2,44}$
$B_{2,45}$	$B_{2,46}$	$B_{2,47}$	$B_{2,48}$	$B_{2,49}$	$B_{2,50}$	$B_{2,51}$	$B_{2,52}$	$B_{2,53}$	$B_{2,54}$	$B_{2,55}$	$B_{2,56}$	$B_{2,57}$	$B_{2,58}$	$B_{2,59}$	$B_{2,60}$
$B_{2,61}$	$B_{2,62}$	$B_{2,63}$	$B_{2,64}$	$B_{2,65}$	$B_{2,66}$	$B_{2,67}$	$B_{2,68}$	$B_{2,69}$	$B_{2,70}$	$B_{2,71}$	$B_{2,72}$	$B_{2,73}$	$B_{2,74}$	$B_{2,75}$	$B_{2,76}$
$B_{2,77}$	$B_{2,78}$	$B_{2,79}$	$B_{2,80}$	$B_{2,81}$	$B_{2,82}$	$B_{2,83}$	$B_{2,84}$	$B_{2,85}$	$B_{2,86}$	$B_{2,87}$	$B_{2,88}$	$B_{2,89}$	$B_{2,90}$	$B_{2,91}$	$B_{2,92}$
$B_{2,93}$	$B_{2,94}$	$B_{2,95}$	$B_{2,96}$	$B_{2,97}$	$B_{2,98}$	$B_{2,99}$	$B_{2,100}$	$B_{2,101}$	$B_{2,102}$	0	0	0	0	0	0

Figure 14: Source block consisting of 3 source UDP packets of lengths 26, 52 and 103 bytes.

8.2.2.8 Void

8.2.2.9 Source FEC Payload ID

The Source FEC payload ID shall be the Source FEC Payload ID format A in section 6.2.2 of RFC6681 [106].

8.2.2.10 Repair FEC payload ID

The Repair FEC Payload ID shall be the Repair FEC Payload ID format A in section 6.2.3 of RFC6681 [106].

8.2.2.10a FEC Object Transmission information

The FEC Object Transmission information consists of:

- the maximum source block length, in symbols
- the symbol size, in bytes

The FEC Object Transmission information shall be the first four octets of the FEC Scheme Specific Information in section 6.2.1.2 of RFC6681 [106].

NOTE: This corresponds to Payload ID Format A in RFC6681 [106] as the last octet of FEC Scheme Specific Information is omitted.

The Source Block Length signalled within the Repair FEC Payload ID of any packet of a stream shall not exceed the Maximum Source Block Length signalled within the FEC Object Transmission Information for the stream.

The FEC Object Transmission Information shall be communicated as described in sub-clause 8.2.2.14. Note, the FEC Object Transmission Information is only communicated in SDP.

8.2.2.11 Hypothetical FEC Decoder

This clause specifies the hypothetical FEC decoder and its use to check packet stream and MBMS receiver conformance.

The hypothetical FEC decoder uses the packet stream, the transmission time of each packet, the initial buffering delay, and the SDP for the stream as inputs. The packet stream from the beginning of the FEC source block until the end of the stream shall comply with the hypothetical reference decoder as specified below when the initial buffer delay equals to the value of the min-buffer-time parameter.

The maximum hypothetical FEC decoding buffer size for MBMS streaming is 1 Mbytes. The default hypothetical FEC decoding buffer size is equal to 1 Mbytes.

For the packet stream, the buffer occupancy level of the hypothetical FEC decoding buffer shall not exceed the value of the buf-size parameter, when it is present in the SDP, or the default FEC decoding buffer size, when the buf-size parameter is not present in the SDP. The output of the hypothetical FEC decoder shall comply with the RTP payload and decoding specifications of the media format.

The hypothetical FEC decoder operates as follows:

1) The hypothetical FEC decoding buffer is initially empty.

2) Each FEC source packet and FEC repair packet, starting from the first packet in transmission order, is inserted into a FEC source block at its transmission time. The FEC source block generation is done as specified in [106], section 6.2.3. The FEC source block resides in the hypothetical FEC decoding buffer.

3) When both the last FEC source packet and the last FEC repair packet of an FEC source block are transmitted, any elements of the FEC source block that are not original UDP packets (e.g. FEC repair packets and potential padding bytes) are removed from the hypothetical FEC decoding buffer.

4) Original UDP packets are not removed from the hypothetical FEC decoding buffer before the signaled initial buffering delay has expired. Then, the first original UDP packet in sequence number order is output and removed from the hypothetical FEC decoding buffer immediately. Each succeeding original UDP packet is output and removed when the following conditions are true:

   i. The following time (in seconds) since the removal of the previous packet has elapsed:

      8 × (size of the previous original UDP packet *including* UDP/IP header in bytes) / (1 000 × (value of "b=AS" SDP attribute for the stream))

   ii. All the packets in the same FEC source block as the original UDP packet have been transmitted.

An MBMS client shall be capable of receiving a packet stream that complies with the hypothetical FEC decoder. Furthermore, in the case of RTP packets, when an MBMS client complies with the requirements for the media decoding of the packet stream, it shall be able to de-packetize and decode the packet stream and output decoded data at the correct rate specified by the RTP timestamps of the received packet stream.

8.2.2.12 Void

8.2.2.13 Signalling

The signalling for streaming FEC consists of several components:

- If several user services are bundled together they are indicated as a sequence of services in the User Service Bundle Description. See sub-clause 11.2.

- A separate SDP describing the FEC repair stream and all the flow IDs referenced from the User Service Bundle Description. See sub-clauses 11.2 and 8.2.2.14.

- SDP protocol identifiers and attributes to indicate the usage of the source packet format, how the FEC payload ID is configured and other FEC parameters such as minimal buffering delay, for the RTP/RTCP streams. See sub-clause 8.2.2.13a.

- Security description extensions to indicate usage of FEC source packet format, and the FEC parameters. See sub-clauses 11.3 and 8.2.2.13a.

The user service description contains either a single service or several bundled services. All of the streaming delivery methods and security descriptions that are present within the *bundleDescription* element must be considered when configuring the FEC operations. This includes RTP, RTCP and MIKEY flows. A receiver intending to perform FEC decoding to cover for packet losses shall receive all the flows that are indicated to be sent as FEC source packets, even if the flows are in a service currently not played out. A receiver intending to use FEC shall also receive the FEC repair stream as described by the FEC Repair Stream Description. The delivery method's session description, and the security description both carry the FEC source packet configuration information: FEC encoding ID, FEC instance ID, and FEC OTI information. The FEC repair packet stream is configured using the similar methods as for the source packets, with the addition of the Flow ID information and buffer delay parameter.

8.2.2.13a SDP for FEC source packet streams

To indicate the presence of the FEC layer between IP/UDP and, RTP or SRTP a SDP protocol identifier is used. Instead of the normal RTP/AVP and RTP/SAVP protocol identifiers, 'UDP/MBMS-FEC/RTP/AVP' and 'UDP/MBMS-FEC/RTP/SAVP' are defined respectively. Both these protocol identifiers shall use the FMT space rules that are used for RTP/AVP and RTP/SAVP respectively, i.e. payload types used in the RTP session is listed. The protocol identifiers are defined in Appendix C1.

The FEC parameters, FEC encoding ID, FEC instance ID and FEC-OTI-Extension information are signalled using the mechanism defined in sub-clause 8.3.1.8. The "a=FEC" SDP attribute shall be used to indicate the single definition that is used for each media component.

For MIKEY messages the service protection description is used to indicate when FEC source packet shall be used, see sub-clause 11.3. The FEC parameter used is also defined in the service protection description. As all MIKEY packets from all user services arrive on the same port, the receiver must use the destination address to separate FEC protected packets from not FEC protected packets. This requires that all MIKEY packets sent to a specific destination address are either FEC protected or not. Note that it is not possible to mix protected and non-protected packets within a single stream as there is no mechanism to determine whether they are protected or not.

8.2.2.14 SDP for FEC repair packet streams

The repair packet stream is indicated in SDP using a media block with the protocol identifier "UDP/MBMS-REPAIR". The media type shall be "application". The FEC parameters, FEC encoding ID, FEC instance ID, FEC-OTI-Extension information and repair parameters (min-buffer-time) are signalled using the mechanisms defined in sub-clause 8.3.1.9. Each media component shall reference only one FEC declaration.

The mapping of the FEC source block flow ID to the destination IP address and UDP port are done using the SDP attribute "a=mbms-flowid" defined in sub-clause 8.3.1.9.

Interleaving may be signaled using the "X-3gpp-FEC-Interleaving" attribute, which also gives the arrangement of the flows in the source block and by consequence their transmission order. The "X-3gpp-FEC-Interleaving" attribute is defined in sub-clause 8.3.1.11.

8.2.2.15 Signalling example for FEC

This sub-clause contains a complete signalling example for a MBMS multicast mode session using FEC with a Service description, a SDP for the streaming delivery method, a SDP for the FEC repair stream, and a security description.

The following is an example bundleDescription.

```
<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription USD-schema-main.xsd"
    fecDescriptionURI="http://www.example.com/3gpp/mbms/session1-fec.sdp">
    <userServiceDescription
        serviceId="urn:3gpp:0010120123hotdog">
            <deliveryMethod
                sessionDescriptionURI="http://www.example.com/3gpp/mbms/session1.sdp"
                protectionDescriptionURI="http://www.example.com/3gpp/mbms/sec-descript">
```

```
                    <sv:delimiter>0</sv:delimiter>
                </deliveryMethod>
                <sv:delimiter>0</sv:delimiter>
        </userServiceDescription>
        <sv:schemaVersion>1</sv:schemaVersion>
    </bundleDescription>
```

The security description has the URI: http://www.example.com/3gpp/mbms/sec-descript

```
    <?xml version="1.0" encoding="UTF-8"?>
    <securityDescription
        xmlns="urn:3GPP:metadata:2005:MBMS:securityDescription"
        xmlns:xsi="”http://www.w3.org/2001/XMLSchema-instance”
        xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:securityDescription security.xsd">
        <keyManagement
            offsetTime="5"
            randomTimePeriod="10">
            <serverURI>http://register.example.com/</serverURI>
            <serverURI>http://register2.example.com/</serverURI>
        </keyManagement>
        <keyId>
            <mediaFlow flowID="FF1E:03AD::7F2E:172A:1E24/4002">
                <MSK>
                    <keyDomainID>aMoM</keyDomainID>
                    <MSKID>aMoAAA==</MSKID>
                </MSK>
            </mediaFlow>
            <mediaFlow flowID="FF1E:03AD::7F2E:172A:1E24/4004">
                <MSK>
                    <keyDomainID>GM8M</keyDomainID>
                    <MSKID>aMkAAA==</MSKID>
                </MSK>
            </mediaFlow>
        </keyId>
        <fecProtection
            fecEncodingId="1"
            fecOtiExtension="ACAEAA=="/>
</securityDescription>
```

An example of how the SDP http://www.example.com/3gpp/mbms/session1.sdp could look for a session containing two media streams that are FEC protected. In this example we have assumed an audiovisual stream, using 56 kbps for video and 12 kbps for audio. In addition another 300 bits/second of RTCP packets from the source is used for the each of the sessions. Hence, the total media session bandwidth is 56+12+0.3+0.3 = 68.6 kbps.

```
v=0
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9
s=3GPP MBMS Streaming SDP Example
i=Example of MBMS streaming SDP file
u=http://www.infoserver.example.com/ae600
e=ghost@mailserver.example.com
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
t=3034423619 3042462419
b=AS:62
b=TIAS: 60500
a=maxprate: 25
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
a=FEC-declaration:0 encoding-id=1
m=video 4002 UDP/MBMS-FEC/RTP/AVP 96
b=TIAS:55000
b=RR:0
b=RS:300
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=framesize:96 176-144
a=FEC:0
a=maxprate:15
m=audio 4004 UDP/MBMS-FEC/RTP/AVP 98
b=TIAS: 11500
b=RR:0
b=RS:300
a=rtpmap:98 AMR/8000
a=fmtp:98 octet-align=1
a=FEC:0
a=maxprate:10
```

The FEC stream used to protect the above RTP sessions and a MIKEY key stream has the below SDP (http://www.example.com/3gpp/mbms/session1-fec.sdp):

```
v=0
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9
s=3GPP MBMS Streaming FEC SDP Example
i=Example of MBMS streaming SDP file
u=http://www.infoserver.example.com/ae600
e=ghost@mailserver.example.com
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
t=3034423619 3042462419
b=AS:15
a=FEC-declaration:0 encoding-id=1
a=FEC-OTI-extension:0 ACAEAA==
a=mbms-repair: 0 min-buffer-time=2600
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
m=application 4006 UDP/MBMS-REPAIR *
b=AS:15
a=FEC:0
a=mbms-flowid: 1=FF1E:03AD::7F2E:172A:1E24/4002, 2=FF1E:03AD::7F2E:172A:1E24/4003,
3=FF1E:03AD::7F2E:172A:1E24/4004, 4=FF1E:03AD::7F2E:172A:1E24/4005, 5=FF1E:03AD::7F2E:172A:1E24/2269
a=X-3gpp-FEC-Interleaving: 1="reverse", 2="ordered"
```

A more traditional FEC configuration is shown below. The audio and video media components use different FEC repair flows. The same principle can also be applied when bundling several user services together.

```
    <?xml version="1.0" encoding="UTF-8"?>
    <bundleDescription
        xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
        xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription USD-schema-main.xsd"
        fecDescriptionURI="http://www.example.com/3gpp/mbms/session2-fec.sdp">
        <userServiceDescription
            serviceId="urn:3gpp:0010120123hotdog">
                <deliveryMethod
                    sessionDescriptionURI="http://www.example.com/3gpp/mbms/session2.sdp">
                    <sv:delimiter>0</sv:delimiter>
               </deliveryMethod>
               <sv:delimiter>0</sv:delimiter>
        </userServiceDescription>
    <sv:schemaVersion>1</sv:schemaVersion>
 </bundleDescription>
```

The SDP file from above is modified to use two different FEC flows.

```
v=0
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9
s=3GPP MBMS Streaming SDP Example
i=Example of MBMS streaming SDP file
u=http://www.infoserver.example.com/ae600
e=ghost@mailserver.example.com
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
t=3034423619 3042462419
b=AS:62
b=TIAS: 60500
a=maxprate: 25
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
m=video 4002 UDP/MBMS-FEC/RTP/AVP 96
b=TIAS:55000
b=RR:0
b=RS:300
a=FEC-declaration:0 encoding-id=1
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=framesize:96 176-144
a=FEC:0
a=maxprate:15
m=audio 4004 UDP/MBMS-FEC/RTP/AVP 98
b=TIAS: 11500
b=RR:0
b=RS:300
a=FEC-declaration:1 encoding-id=1
```

```
a=rtpmap:98 AMR/8000
a=fmtp:98 octet-align=1
a=FEC:1
a=maxprate:10
```

The SDP file for the two FEC streams

```
v=0
o=ghost 2890844526 2890842807 IN IP6 2001:210:1:2:240:96FF:FE25:8EC9
s=3GPP MBMS Streaming FEC SDP Example
i=Example of MBMS streaming SDP file
u=http://www.infoserver.example.com/ae600
e=ghost@mailserver.example.com
t=3034423619 3042462419
b=AS:15
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
m=application 4006 UDP/MBMS-REPAIR *
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
b=AS:15
a=FEC-declaration:0 encoding-id=1
a=FEC-OTI-extension:0 ACAEAA==
a=mbms-repair: 0 min-buffer-time=2600
a=FEC:0
a=mbms-flowid: 1=FF1E:03AD::7F2E:172A:1E24/4002, 2=FF1E:03AD::7F2E:172A:1E24/4003
m=application 4008 UDP/MBMS-REPAIR *
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
b=AS:15
a=FEC-declaration:1 encoding-id=1
a=FEC-OTI-extension:1 ACAEAA==
a=mbms-repair: 1 min-buffer-time=2600
a=FEC:1
a=mbms-flowid: 3=FF1E:03AD::7F2E:172A:1E24/4004, 4=FF1E:03AD::7F2E:172A:1E24/4005
```

8.2.3 General RTP Header Extension Mechanism

8.2.3.1 Introduction

The General RTP Header Extension Mechanism [92] is a general mechanism to use the header extension feature of RTP (the Real-Time Transport Protocol). The General RTP Header Extension Mechanism should be supported.

8.2.3.2 Timestamp Offset

Timestamp offsets for RTP may be transmitted using the general RTP header extension mechanism.

The variable timestamp extension element is 32 bits long. The first byte is the extension element header, i.e. the ID and len fields, as defined in [92]. The remaining 3 bytes are the timestamp-offset measured in the same frequency as the RTP timestamp.

Timestamp-offset: A 24 bit unsigned integer signaling the offset of the received packets of the same media in the tune-in FEC block. The timestamp offset indicates at most the difference between the RTP timestamp of the current packet and the highest RTP timestamp of packets of the same media stream that are transmitted in the current FEC source block.

Timestamp offset shall not be used if FEC protection and Interleaving are not being used.

The following example is a general RTP header extension block containing a single variable timestamp extension element.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          0xBEDE               |           length=1            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  ID   | len=2 |             timestamp-offset                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The presence of variable timestamps is signaled in the SDP file using the header extension specification and the URI "http://www.3gpp.org/2008/TimestampOffset". The URI signals the possible presence of timestamp offsets with the given ID.

8.3 Session description

SDP is provided to the MBMS client via a discovery/announcement procedure to describe the streaming delivery session. The SDP describes one or more RTP sessions part of the MBMS streaming session. The SDP shall be a correctly formed SDP according to [14].

8.3.1 SDP Parameters for MBMS streaming session

The semantics of a Session Description of an MBMS streaming session shall include the parameters:

- The sender IP address.
- The number of media in the session.
- The destination IP address and port number for each and all of the RTP sessions in the MBMS streaming session.
- The start time and end time of the session.
- The protocol ID (i.e. RTP/AVP).
- Media type(s) and fmt-list.
- Data rate using existing SDP bandwidth modifiers.
- Mode of MBMS bearer per media.
- FEC configuration and related parameters.
- Service-language(s) per media.
- QoE Metrics (defined in sub-clauses 8.3.2.1 and 8.4).

8.3.1.1 Sender IP address

There shall be exactly one IP source address per media description within the SDP. The IP source address shall be defined according to the source-filter attribute ("a=source-filter:") [15] 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 MBMS streaming 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.

8.3.1.2 Destination IP address and port number for channels

Each RTP session part of a MBMS streaming session is defined by two parameters:

- IP destination address.
- Destination port number(s).

The IP destination address shall be defined according to the "connection data" field ("c=") of [14]. The destination port number shall be defined according to the <port> sub-field of the media announcement field ("m=") of [14]. Multiple ports using "/" notation shall not be used. The RTCP port, if used, shall be RTP port +1.

8.3.1.3 Media Description

The media description line shall be used as defined in [14] for RTP. The <media> part indicates the type of media, audio, video, or text. The usage of RTP and any applicable RTP profile shall be indicated by using the <proto> field of the '*m*-line'. The one or more payload types that are being used in this RTP session are enumerated in the <fmt> part. Each payload type is declared using the "a=rtpmap" attribute according to [14] and use the "a=fmtp" line when required to describe the payload format parameters.

8.3.1.4 Session Timing Parameters

A MBMS streaming session start and end times shall be defined according to the SDP timing field ("t=") - [14].

8.3.1.5 Mode of MBMS bearer per media

The MBMS bearer mode declaration attribute shall be used for MBMS streaming sessions, as defined in sub-clause 7.3.2.7.

8.3.1.6 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 [74] and ISO 3166 [75] (e.g. "a=lang:EN-US"). These are the same tags used in the User Service Description XML.

8.3.1.7 Bandwidth specification

The bit-rate required by the MBMS streaming session and its media components shall be specified using both the "AS" bandwidth modifier and the "TIAS" bandwidth modifier combined with "a=maxprate" [38] on media level in the SDP. On session level the "TIAS" bandwidth modifier combined with "a=maxprate" may be used, where the session level expresses the aggregated peak bit-rate, which may be lower than the sum of the individual media streams.

The bandwidth required for RTCP is specified by the "RR" and "RS" bandwidth modifiers (3GPP TS 26.244 [32]) on media level for each RTP session. The "RR" modifier shall be included and set to 0 to specify that RTCP receiver reports are not used. The bandwidth used for RTCP sender reports shall be specified using the "RS" bandwidth modifier.

8.3.1.8 FEC Parameters

The FEC encoding ID and instance ID are provided using the "a=FEC-declaration" attribute defined in sub-clause 7.3.2.8. Any OTI information for that FEC encoding ID and instance ID is provided with below defined FEC OTI attribute.

The FEC OTI attribute must be immediately preceded by the "a=FEC-declaration" attribute (and so can be session-level and media-level). The fec-ref maps the oti-extension to the FEC-declaration OTI it extends. The purpose of the oti-extension is to define FEC code specific OTI required for RTP receiver FEC payload configuration; exact contents are FEC code specific and need to be specified by each FEC code using this attribute. The OTI for the MBMS FEC Scheme is defined in sub-clause 8.2.2.10a.

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

- sdp-fec-oti-extension-line = "a=FEC-OTI-extension:" fec-ref SP oti-extension CRLF
- fec-ref = 1*3DIGIT (the SDP-internal identifier for the associated FEC-declaration).
- oti-extension = base64
- base64 = *base64-unit [base64-pad]

- base64-unit = 4base64-char
- base64-pad = 2base64-char "==" / 3base64-char "="
- base64-char = ALPHA / DIGIT / "+" / "/"

To provide the FEC repair packets with additional, non FEC specific parameters, a session and media level SDP attribute is defined.

sdp-fec-parameter-line = “a=mbms-repair: 0*1SP fec-ref SP parameter-list CRLF

parameter-list = parameter-spec *(1*SP parameter-spec)

parameter-spec = name “=” value;

name = 1*(ALPHA / DIGIT / “-”)

value = 1*(safe) ; safe defined in [14]

Currently one FEC non code-specific parameter is defined:

min-buffer-time: This FEC buffering parameter specifies the minimum receiver buffer time (delay) needed to ensure that FEC repair has time to happen regardless of the FEC source block of the stream from which the reception starts. The value is in milliseconds and represents the wallclock time between the reception of the first FEC source or repair packet of a FEC source block, whichever is earlier in transmission order, and the wallclock time when media decoding can safely start.

The parameters name and value is defined in ABNF as follows:

Min-buffer-time-parameter-name = “min-buffer-time”

Min-buffer-time-parameter-value = 1*8DIGIT ;Wallclock time in milliseconds.

The FEC declaration and FEC OTI information utilized in a specific source or repair packet is indicated using the FEC-ref number in the a=fec lines as described in sub-clauses 8.2.2.12 and 8.2.2.13.

8.3.1.9 FEC Flow ID attribute

To indicate the mapping between destination IP address and UDP port number and FEC source block flow IDs, the “a=mbms-flowid” SDP attribute is defined. Each flowID that is used to construct a source block within the bundled sessions shall be included. It is a media level attribute that shall be present in any SDP media block using the “UDP/MBMS-REPAIR” protocol identifier.

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

Sdp-mbms-flowid-attr = "a=mbms-flowid:" *WSP flow-id-spec *("," *WSP flow-id-spec) CRLF

flow-id-spec = flowID "=" address-spec "/" port-spec

address-spec = IP4-multicast / IP6-multicast

IP4-multicast = m1 3*("." decimal-uchar)

m1 = ("22" ("4"/"5"/"6"/"7"/"8"/"9")) / ("23" DIGIT))

IP6-multicast = hexpart

hexpart = hexseq / hexseq "::" [hexseq] /

"::" [hexseq]

hexseq = hex4 *(":" hex4)

hex4 = 1*4HEXDIG

port-spec = 1*5DIGIT

8.3.1.10 Buffer Requirement Signaling

Due to the variable bitrate nature of some media streams (especially video streams), initial buffering at the receiver becomes necessary to smooth out those variations. The initial buffering delay SHOULD be signaled to the receiver in the SDP using the following media level attribute:

- "a=X-initpredecbufperiod:<initial pre-decoder buffering period>"

For H.263 video streams, the “X-initpredecbufperiod” indicates the required initial pre-decoder buffering period specified according to Annex G of 3GPP TS 26.234 [47].

For H.264 video streams, the “X-initpredecbufperiod” indicates the nominal removal time of the first access unit from the coded picture buffer (CPB).

Note that X-initpredecbufperiod is expressed as clock ticks of a 90-kHz clock. Hence, conversion may be required if the RTP timestamp clock frequency is not 90 kHz.

8.3.1.11 Interleaving Signaling

When interleaving is used in combination with FEC protection of an MBMS service, the BM-SC may indicate to receivers the order of transmission of the media units of a source block using the “X-3gpp-FEC-Interleaving” attribute. It also indicates whether intra-stream interleaving (described in section G.1) has been performed or not for each of the flows in the FEC source block.

The “X-3gpp-FEC-Interleaving” attribute is defined as follows:

Interleaving=”X-3gpp-FEC-Interleaving:” SP flow_interleaving *(“,” flow_interleaving) CRLF

flow_interleaving=flowID “=” [“ordered” / “mixed” / ”reverse”]

flowID is the indentification of the flow as described in section 8.3.1.9. The intra-stream interleaving modes may result in un-changed tranmission order (“Ordered”), a mixed transmission order (“Mixed”), or a reversed transmission order (“Reverse”). For a flow that is not listed in the X-3gpp-FEC-Interleaving attribute, the receiver should assume that no particular intra- or inter-stream interleaving has been performed. The transmission order does not preculde that some media units of a lower priority stream are interleaved with the media units of higher priority stream.

8.3.2 SDP Example for Streaming Session

Here is a full example of SDP description describing the media streams part of a streaming delivery session:

```
v=0
o=ghost 2890844526 2890842807 IN IP4 192.168.10.10
s=3GPP MBMS Streaming SDP Example
i=Example of MBMS streaming SDP file
u=http://www.infoserver.example.com/ae600
e=ghost@mailserver.example.com
c=IN IP6 FF1E:03AD::7F2E:172A:1E24
t=3034423619 3042462419
b=AS:77
a=mbms-mode:broadcast 123869108302929 1
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
m=video 4002 RTP/AVP 96
b=TIAS:62000
b=RR:0
b=RS:600
a=maxprate:17
a=rtpmap:96 H264/90000
a=fmtp:96 profile-level-id=42A01E; packetization-mode=1; sprop-parameter-sets=Z0IACpZTBYmI,aMljiA==
m=audio 4004 RTP/AVP 98
b=TIAS:15120
b=RR:0
b=RS:600
a=maxprate:10
a=rtpmap:98 AMR/8000
a=fmtp:98 octet-align=1
```

FEC is not used in that example. See clause 8.2.2.15 for an example with FEC.

8.3.2.1 SDP Description for QoE Metrics

Similar as in 3GPP TS 26.234 [47], an SDP attribute for QoE, which can be used either at session or media level, is defined below in [23] based on [14]:

- QoE-Metrics-line = "a" "=" "3GPP-QoE-Metrics:" att-measure-spec *("," att-measure-spec)) CRLF
- att-measure-spec = Metrics ";" Sending-rate [";" Measure-Range] [";" Measure-Resolution] *([";" Parameter-Ext])
- Metrics = "metrics" "=" "{"Metrics-Name *("|" Metrics-Name) " }"
- Metrics-Name = 1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7e) ;VCHAR except ";", ",", "{" or "}"
- Sending-Rate = "rate" "=" 1*DIGIT / "End"
- Measure-Resolution = "resolution" "=" 1*DIGIT ; in seconds
- Measure-Range = "range" ":" Ranges-Specifier
- Parameter-Ext = (1*DIGIT ["." 1*DIGIT]) / (1*((0x21..0x2b) / (0x2d..0x3a) / (0x3c..0x7a) / 0x7c / 0x7e))
- Ranges-Specifier = as defined in RFC 2326 [88].

An MBMS server uses this attribute to indicate that QoE metrics are supported and shall be used if also supported by the MBMS client. When present at session level, it shall only contain metrics that apply to the complete session. When present at media level, it shall only contain metrics that are applicable to individual media.

The "Metrics" field contains the list of names that describes the metrics/measurements that are required to be reported in a MBMS session (see sub-clause 8.4). The names that are not included in the "Metrics" field shall not be reported during the session.

In this version of the specification, the "Sending-Rate" shall be set to the value "End", which indicates that only one report is sent at the end of the MBMS session.

The optional "Measure-Resolution" field, if used, shall define a time over which each metrics value is calculated. The "Measure-Resolution" field splits the session duration into a number of equally sized periods where each period is of the length specified by the "Measure-Resolution" field. The "Measure-Resolution" field is thus defining the time before the calculation of a QoE parameter starts over. If the "Measure-Resolution" field is not present the metrics resolution shall cover the period specified by the "Measure-Range" field. If the "Measure-Range" field is not present the metrics resolution shall be the whole session duration.

The "Measure-Resolution" field shall take only one value for all session level metrics and only one value for all metrics associated to one media. Note that "Measure-Resolution" shall be evaluated according to a real-time clock. This implies that the real-time interval between consecutive measurements is not affected by changes in playback rate, for instance due to buffering.

The optional "Measure-Range" field, if used, shall define the time range in the stream for which the QoE metrics will be reported. There shall be only one range per measurement specification. The range format shall be any of the formats allowed by the media. If the "Measure-Range" field is not present, the corresponding (media or session level) range attribute in SDP shall be used. If SDP information is not present, the metrics range shall be the whole session duration.

8.3.2.2 OMA-DM Configuration of QoE Metrics

As a supplement to QoE provisioning per session (as specified in 8.3.2.1), OMA-DM can be used to specify QoE configuration. If such QoE configuration has been specified, it should be used by the terminal for all subsequent MBMS sessions. Note that the use of OMA-DM for configuring QoE reporting is applicable to either streaming or download delivery sessions over which streaming services are delivered.

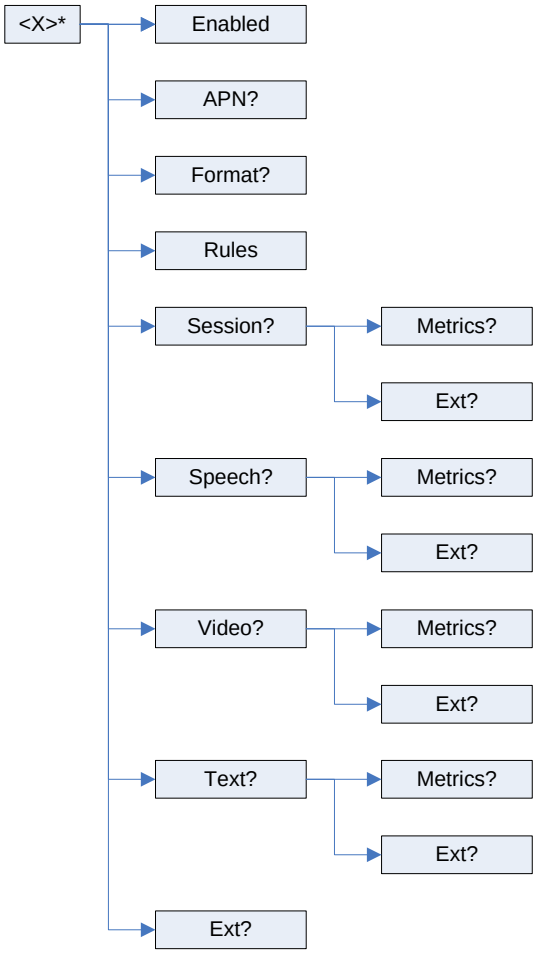
From the QoE reporting perspective, session-specific and OMA-DM provisioned QoE configuration shall be considered separate and independent processes. A UE shall consider both these types of reception reporting to be valid, if it receives QoE configuration parameters for both of them.

For OMA-DM QoE configuration the parameters are specified according to the following Managed Object (MO). Version numbering is included for possible extension of the MO.

The Management Object Identifier shall be: urn:oma:mo:ext-3gpp-mbmsqoe:1.0.

Protocol compatibility: The MO is compatible with OMA Device Management protocol specifications, version 1.2 and upwards, and is defined using the OMA DM Device Description Framework as described in the Enabler Release Definition OMA-ERELD _DM-V1_2 [94].

The following nodes and leaf objects shall be contained under the 3GPP_MBMSQOE node if an MBMS client supports the feature described in this clause (information of DDF for this MO is given in Annex H):



Node: /<X>

This interior node specifies the unique object id of a MBMS QoE metrics management object. The purpose of this interior node is to group together the parameters of a single object.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

The following interior nodes shall be contained if the MBMS client supports the “MBMS QoE metrics Management Object”.

/<X>/Enabled

This leaf indicates if QoE reporting is requested by the provider.

- Occurrence: One
- Format: bool
- Minimum Access Types: Get

/<X>/APN

This leaf contains the Access Point Name that should be used for establishing the PDP context on which the QoE metric reports will be transmitted. This may be used to ensure that no costs are charged for QoE metrics reporting. If this leaf is not defined then any QoE reporting is done over the access point according to sub-clause 11.2.1.1.

- Occurrence: ZeroOrOne
- Format: chr
- Minimum Access Types: Get
- Values: the Access Point Name

/<X>/Format

This leaf specifies the format of the report and if compression (Gzip XML) is used.

- Occurrence: ZeroOrOne
- Format: chr
- Minimum Access Types: Get
- Values: “XML”, “GZIPXML”.

/<X>/Rules

This leaf provides in XML format the rules used to decide if and how the reports are sent to the QoE metrics report server. The leaf also provides the URIs of one or more servers which shall be the receiver of the QoE metrics report. In case of multiple servers, the MBMS client randomly selects one of the servers from the list, with uniform distribution.

The XML scheme is described in sub-clause 9.5.1 and an example XML code is shown in sub-clause 9.5.2. Only the *postReceptionReport* part needs to be specified. The use of OMA-DM based configuration of QoE reception reporting shall be independent of any such configuration specified by session-specific mechanisms such as SDP description or Associated Delivery Procedure Description. In other words, QoE reception reporting procedure by the UE as determined by parameters set by OMA-DM configuration shall occur independently and separately from its reporting procedure as determined by parameters set in the Associated Delivery Procedures Description or the SDP description, and vice versa.

- Occurrence: One
- Format: chr
- Minimum Access Types: Get
- Values: See clause Annex H.

/<X>/Ext

The Ext node is an interior node where the vendor specific information can be placed (vendor includes application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Session

The Session node is the starting point of the session level QoE metrics definitions.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Session/Metrics

This leaf provides in textual format the QoE metrics that need to be reported, the measurement frequency, the reporting interval and the reporting range. The syntax and semantics of this leaf are defined in clause 8.3.2.1.

- Occurrence: ZeroOrOne
- Format: chr
- Minimum Access Types: Get
- Values: see clause 8.3.2.1.

/<X>/Session/Ext

The Ext node is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Speech

The Speech node is the starting point of the speech/audio media level QoE metrics definitions.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Speech/Metrics

This leaf provides in textual format the QoE metrics that need to be reported, the measurement frequency, the reporting interval and the reporting range. The syntax and semantics of this leaf are defined in clause 8.3.2.1.

- Occurrence: ZeroOrOne
- Format: chr
- Minimum Access Types: Get
- Values: see clause 8.3.2.1.

/<X>/Speech/Ext

The Ext node is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Video

The Video node is the starting point of the video media level QoE metrics definitions.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Video/Metrics

This leaf provides in textual format the QoE metrics that need to be reported, the measurement frequency, the reporting interval and the reporting range. The syntax and semantics of this leaf are defined in clause 8.3.2.1.

- Occurrence: ZeroOrOne
- Format: chr
- Access Types: Get
- Values: see clause 8.3.2.1.

/<X>/Video/Ext

The Ext is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the Ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

/<X>/Text

The Text node is the starting point of the timed-text media level QoE metrics definitions.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get
- Values: see clause 8.3.2.1.

/<X>/Text/Metrics

This leaf provides in textual format the QoE metrics that need to be reported, the measurement frequency, the reporting interval and the reporting range. The syntax and semantics of this leaf are defined in clause 8.3.2.1.

- Occurrence: ZeroOrOne
- Format: chr
- Minimum Access Types: Get
- Values: see clause 8.3.2.1.

/<X>/Text/Ext

The Ext is an interior node where the vendor specific information can be placed (vendor meaning application vendor, device vendor etc.). Usually the vendor extension is identified by vendor specific name under the ext node. The tree structure under the vendor identified is not defined and can therefore include one or more un-standardized sub-trees.

- Occurrence: ZeroOrOne
- Format: node
- Minimum Access Types: Get

8.4 Quality of Experience

8.4.1 General

The MBMS Quality of Experience (QoE) metrics feature is optional for both MBMS streaming server and MBMS client, and shall not disturb the MBMS service. An MBMS Server that supports the QoE metrics feature shall activate the gathering of client QoE metrics with SDP as described in sub-clauses 8.3.2.1 and 8.4.2 and via the reception reporting procedure as described in sub-clause 9.4. Alternatively QoE activation can be done with OMA-DM as described in sub-clause 8.3.2.2. An MBMS client supporting the feature shall perform the quality measurements in accordance to the measurement definitions, aggregate them into client QoE metrics and report the metrics to the specified server using the content reception reporting procedure. The way the QoE metrics are processed and made available is out of the scope of the present document.

8.4.2 QoE Metrics

An MBMS client should measure the metrics at the transport layer after FEC decoding (if FEC is used), but may also do it at the application layer for better accuracy.

The measurement period for the metrics is the whole streaming duration and the measurement resolution of each reported metrics value is defined by the "Measure-Resolution" field. The measurement period may be less than the session duration, because of late joiners or early leavers. The measurement period shall not include any voluntary event that impacts the actual play, such as pause, or any buffering or freezes/gaps caused by them.

The following metrics in Table 8.4.2 shall be derived by the MBMS client implementing QoE:

Table 8.4.2

QoE Metric	Streaming delivery method	Download delivery method	Metric type
Corruption duration metric	✓		Media
Rebuffering duration metric	✓		Session
Initial buffering duration metric	✓		Session
Successive loss of RTP packets	✓		Media
Frame rate deviation	✓		Media
Jitter duration	✓		Media
Content Access/Switch Time	✓		Session
Network Resource	✓	✓	Session

Average codec bitrate	✓		Media
Codec information	✓		Media
Loss of Objects[1]		✓	Session
Distribution of Symbol Count Underrun for Failed Blocks[1]		✓	Session

[1] These metrics are of interest mainly for sessions with a large number of object deliveries such as HTTP streaming sessions [98].

All media metrics are only applicable to at least one of audio, video, speech and timed text media types, and are not applicable to other media types such as synthetic audio, still images, bitmap graphics, vector graphics, and text.

Any unknown metrics shall be ignored by the client and not included in any QoE report.

8.4.2.1 Corruption duration metric

Corruption duration, M, is the time period from the NPT time of the last good frame before the corruption, (since the NPT time for the first corrupted frame cannot always be determined) or the start of the measurement period (whichever is later)to the NPT time of the first subsequent good frame or the end of the measurement period (whichever is sooner). A corrupted frame may either be an entirely lost frame, or a media frame that has quality degradation and the decoded frame is not the same as in error-free decoding. A good frame is a "completely received" frame X that, either:

- it is a refresh frame (does not reference any previously decoded frames AND where none of the subsequently decoded frames reference any frames decoded prior to X); or
- does not reference any previously decoded frames; or
- only references previously decoded "good frames".

"Completely received" means that all the bits are received and no bit error has occurred.

Corruption duration, M, in milliseconds can be calculated according to the derivation of good frames as below:

a) A good frame can be derived by the client using the codec layer, in which case the codec layer signals the decoding of a good frame to the client. A good frame could also be derived by error tracking methods, but decoding quality evaluation methods shall not be used. An error tracking method may derive that a frame is a good frame even when it references previously decoded corrupted frames, as long as all the referenced pixels for generating the prediction signal were correctly reconstructed when decoding the reference frames. A decoding quality evaluation method may derive that a frame is a good frame even one or more pixels of the frame have not been correctly reconstructed, as long as the decoding quality is considered by the method as acceptable. Such a frame is not a good frame according to the definition above, which shall be strictly followed.

b) In the absence of information from the codec layer, a good frame should be derived according to N, where N is optionally signalled from MBMS streaming server (via SDP) to the MBMS client and represents the maximum duration, in presentation time, between two subsequent refresh frames in milliseconds. After a corrupted frame, if all subsequent frames within N milliseconds in presentation time have been completely received, then the next frame is a good frame.

c) N is not signalled, then it defaults to ∞ (for video) or to one frame duration (for audio).

The optional parameter D is defined to indicate which of options a) and b) is in use. D is signalled from the client to the server. When D is equal to "a", option a) shall be in use, and the optional parameter T shall be present. When D is equal to "b", option b) shall be in use and the optional parameter T shall not be present.

The optional parameter N as defined in point b is used with the "Corruption_Duration" parameter. The optional parameter T is defined to indicate whether the client uses error tracking (when T is equal to "On") or not (when T is equal to "Off"). T is signalled from the client to the server.

The syntax for D, N to be included in the "att-measure-spec" (sub-clause 8.3.2.1) is as follows:

- D = "D" "=" "a" / "b"

- N = "N" "=" 1*DIGIT

In MBMS reception reporting will be done only once at the end of streaming, hence all the occurred corruption durations are summed up over each resolution period of the stream and stored in the vector *TotalCorruptionDuration*. The unit of this metrics is expressed in milliseconds. For each resolution duration the number of individual corruption events are summed up and stored in the vector *NumberOfCorruptionEvents*. These two vectors are reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.2 Rebuffering duration metric

Rebuffering is defined as any stall in playback time due to any involuntary event at the client side.

The syntax for the metric "Rebuffering_Duration" for the QoE-Feedback header is as defined in sub-clause 8.3.2.1.

Rebuffering starts at the NPT time of the last played frame before the occurrence of the rebuffering.

In MBMS reception reporting will be done only once at the end of streaming, hence all the occurred rebuffering durations are summed up over each resolution period of the stream and stored in the vector *TotalRebufferingDuration*. The unit of this metrics is expressed in seconds, and can be a fractional value. The number of individual rebuffering events for each resolution duration are summed up and stored in the vector *NumberOfRebufferingEvents*. These two vectors are reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.3 Initial buffering duration metric

Initial buffering duration is the time from receiving the first RTP packet until playing starts.

The syntax for the "Initial_Buffering_Duration" is as defined in sub-clause 8.3.2.1.

The metric value indicates the initial buffering duration where the unit of this metrics is expressed in seconds, and can be a fractional value. There can be only one measure and it can only take one value. "Initial_Buffering_Duration" is a session level parameter.This value is reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.4 Successive loss of RTP packets

The metric "Successive_Loss" indicates the number of RTP packets lost in succession (excluding FEC packets) per media channel.

The syntax for the metrics "Successive_Loss" is as defined in sub-clause 8.3.2.1.

In MBMS reception reporting will be done only once at the end of streaming, hence all the number of successively lost RTP packets are summed up over each resolution period of the stream and stored in the vector *TotalNumberofSuccessivePacketLoss*. The unit of this metric is expressed as an integer equal to or larger than 0. The number of individual successive packet loss events over each resolution duration are summed up and stored in the vector *NumberOfSuccessiveLossEvents*. The number of received packets is also summed up over each resolution duration and stored in the vector *NumberOfReceivedPackets*. These three vectors are reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.5 Frame rate deviation

Frame rate and frame rate deviation indicates the playback frame rate information. Frame rate deviation happens when the actual playback frame rate during a measurement period is deviated from a pre-defined value.

The actual playback frame rate is equal to the number of frames played during the resolution period divided by the time duration, in seconds, of the actual measurement. For the last measurement period in the session this time duration might be shorter than the configured measurement resolution (see 8.3.2.1 for the definition of the measurement resolution).

The parameter FR that denotes the pre-defined frame rate value is used with the "Framerate_Deviation" parameter in the "3GPP-QoE-Metrics" attribute. The value of FR shall be set by the server. The syntax for FR to be included in the "att-measure-spec" (sub-clause 8.3.2.1) is as follows:

- FR = "FR" "=" 1*DIGIT "." 1*DIGIT

The syntax for the metrics"Framerate" and "Framerate_Deviation" is defined in sub-clause 8.3.2.1

The metric "Framerate" indicates the actual playback frame rate. It is expressed in frames per second, and can be a fractional value..

For the Metrics-Name "Framerate_Deviation", the value field indicates the frame rate deviation value that is equal to the pre-defined frame rate minus the actual playback frame rate. This metric is expressed in frames per second, and can be a fractional value, and can be negative.

The frame rate and the frame rate deviations for each resolution period are stored in the vectors *Framerate* and *FramerateDeviation* and the vectors are reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.6 Jitter duration

Jitter happens when the absolute difference between the actual playback time and the expected playback time is larger than a pre-defined value, which is 100 milliseconds. The expected time of a frame is equal to the actual playback time of the last played frame plus the difference between the NPT time of the frame and the NPT time of the last played frame.

The syntax for the metric "Jitter_Duration" is defined in sub-clause 8.3.2.1.

In MBMS reception reporting will be done only once at the end of streaming, hence all the Jitter_Durations are summed up over each resolution duration and stored in the vector *TotalJitterDuration*. The unit of this metrics is expressed in seconds, and can be a fractional value. The number of individual events over the resolution duration are summed up and stored in the vector *NumberOfJitterEvents*. These two vectors are reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.7 Content Access/Switch Time

Content access/switch time is the time that elapses between the initiation of a content request/switch by the user and up to the time when the first packet of the content or media stream is received.

The syntax for the metric "Content_Access_Time" is defined in sub-clause 8.3.2.1.

The metric value indicates the content access/switch time and the unit of this metrics is expressed in seconds, and can be a fractional value. There can be only one measure and it can only take one value. "Content_Access_Time" is a session level parameter. This value is reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.8 Network Resource

The Network_Resource identifies the cell which has been used during each measurement period.

The syntax for the metric "Network_Resource" is as defined in sub-clause 8.3.2.1.

In GERAN and UTRAN, the cell is identified by the Cell Global Identity (as described in 3GPP TS 23.003 [77]), which is a concatenation of MCC, MNC, LAC and CI. It shall be coded as a text string as follows: Starting with the most significant bit, MCC (3 digits), MNC (2 or 3 digits depending on MCC value), LAC (4 hexadecimal digits) and CI (4 hexadecimal digits).

In E-UTRAN, the cell is identified by the E-UTRAN Cell Global Identification (ECGI) (as described in 3GPP TS 36.331 [97]) which is a concatenation of the PLMN Identifier (PLMN-Id) and the E-UTRAN Cell Identity (ECI). The PLMN identifier consists of MCC and MNC. It shall be coded as a text string as follows: starting with the most significant bit, MCC (3 digits), MNC (2 or 3 digits depending on MCC value) and ECI (7 hexadecimal digits).

Only one cell shall be reported per measurement period, even if more than one cell has been used during a measurement period or if reception is done simultaneously from several cells.

The cell used is stored in the vector *NetworkResourceCellId*. If the value in the vector for a resolution period is unchanged from the previous value, it is allowed to put the value “=” in the vector to indicate this. The vector is reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.9 Average codec bitrate

The average codec bitrate is the bitrate used for coding “active” media information during the measurement resolution period.

For audio media “active” information is defined by frames containing audio. If the audio codec uses silence frames (SID-frames), these frames are not counted as "active", and the SID-frames and the corresponding DTX time periods are excluded from the calculation. Thus for audio media the average codec bitrate can be calculated as the number of audio bits received for “active” frames , divided by the total time, in seconds, covered by these frames. The total time covered is calculated as the number of “active” frames times the length of each audio frame.

For non-audio media the average codec bitrate is the total number of media bits played out during the measurement resolution period, divided by the length of the playout period. The playout period length is normally equal to the length of the measurement resolution period, but if rebuffering occurs the playout period will be shorter (i.e. any rebuffering time shall be ignored when calculating the codec bitrate).

The syntax for the metric "Average_Codec_Bitrate" is defined in sub-clause 8.3.2.1.

The average codec bitrate value for each measurement resolution period shall be stored in the vector *AverageCodecBitrate*. The unit of this metrics is expressed in kbit/s and can be a fractional value. The vector is reported by the client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.10 Codec information

The codec information metrics contain details of the media codec used during the measurement resolution period. If the codec information is changed during the measurement resolution period, the codec information valid when each measurement resolution period ends shall be reported. The unit of this metric is a string value. No "white space" characters are allowed in the string values, and shall be removed if necessary.

For audio media the codec information contains the audio codec type, represented as in an SDP offer, for instance "AMR-WB/16000/1".

For video media, the codec information contains the video codec type, represented as in an SDP offer, for instance “H263-2000/90000”. Furthermore, the video profile and level used, as well as the image size used shall be reported. For instance "profile=0;level=45" for the profile and level information and “176x144” for the image size. In some cases the profile and level is reported together, for instance "profile-level-id=42e00a". Note that the image size reported for each measurement resolution period shall be the one actually used, not the maximum size allowed by the SDP negotiation.

For timed text media, the codec information contains the text encoding, represented as in an SDP offer, for instance "3gpp-tt/1000".

The syntax for the metric "Codec_Info", “Codec_ProfileLevel” and “Codec_ImageSize” are defined in sub-clause 8.3.2.1.

The codec info, profile / level and codec image size value for each measurement resolution period shall be stored in the vectors C*odecInfo*, *CodecProfileLevel* and *CodecImageSize* respectively. If the metric values in these vectors for a measurement resolution period are unchanged from the previous values in the respective vector, it is allowed to put the value “=” in the vector to indicate this. The C*odecInfo, CodecProfileLevel* and *CodecImageSize* vectors are reported by the client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.11 Loss of Objects

The metric "Object_Loss" indicates the number of objects lost in a FLUTE session during a resolution period.

The syntax for the metric "Object_Loss" is as defined in sub-clause 8.3.2.1.

The number of lost objects are summed up over each resolution period of the session and stored in the vector *numberOfLostObjects*. The unit of this metric is expressed as an integer equal to or larger than 0. The number of received objects is also summed up over each resolution duration and stored in the vector *NumberOfReceivedObjects*. These two vectors are reported by the MBMS client as part of the reception report (sub-clauses 9.4.6 and 9.5.3).

8.4.2.12 Distribution of Symbol Count Underrun for Failed Blocks

The elements of the distribution of the metric "Distribution_of_Symbol_Count Underrun" are calculated by subtracting the total number of source symbols, from the number of received symbols for a failed block in a failed object. The range of values of the distribution are limited to the range of interest through top and bottom range parameters. Values greater than the top of the range are reported as the maximum value. Values lower than the bottom of the range are reported as the minimum value.

Reported values may also be grouped in bins. The size of the bins used for collectring statistics are specified through a bin size parameter. The first bin starts at the bottom of the range. The last bin must include the top of the range. Collection bins are adjacent.

The range of file sizes considered for calculating the metric can also be restricted through optional minmum, and maximum file size parameters.

The distribution is reported per measurement duration as a string list of (bin lower bound, number of occurrences) pairs. The bin lower bound uniquely identifies a bin by providing the lowest value of the range of each bin. The bin lower bound, and number of occurrences are both integer. The vector *SymbolCountUnderrun* is a string vector where every entry is a bracket delimited list of (bin lower bound,number of occurrences) pairs. Bins with 0 occurences are omitted from the list. Values greater than the top of the range are reported within the bin containing the top of the range. Values lower than the bottom of the range are reported within the first bin containing the bottom of the range.

The top, bottom, and bin size of the distribution range are provided through the optional parameters T, B, and S respectively. The syntax for T, B, and S to be included in the "att-measure-spec" (sub-clause 8.3.2.1) is as follows:

* T = "T" "=" {+|-} 1*DIGIT
* B = "B" "=" {+|-} 1*DIGIT
* S = “S” “=” 1*DIGIT

The default value of the top of the range, in case the T parameter is omitted, is 0. The default value of the bottom of the range, in case the B parameter is omitted, is -10. The default value of the bin size is 1.

The minimum, and maximum file sizes considered for calculating the metric are provided through the optional parameters Y, and Z respectively. The syntax for Y, and Z to be included in the "att-measure-spec" (sub-clause 8.3.2.1) is as follows:

* Y= "Y" "=" 1*DIGIT
* Z = "Z" "=" 1*DIGIT

The default value of the minimum file size is 0, and the default value of the maximum file size is infinity.

8.4.3 Example metrics initiation with SDP

This following example shows the syntax of the SDP attribute for QoE metrics. The session level QoE metrics description (Initial buffering duration, rebufferings and network resource) are to be monitored with a measurement resolution of 20 seconds and reported at the end of the session. Also video specific description of metrics (corruptions) are to be monitored and reported at the end from the beginning of the stream until the time 40s. Finally, audio specific description of metrics (corruptions) is to be monitored with a measurement resolution of 10s and reported at the end of the stream.

SDP example:

```
v=0
o=- 3268077682 433392265 IN IP4 63.108.142.6
s=QoE Enabled Session Description Example
e=support@foo.com
c=IN IP4 0.0.0.0
t=0 0
a=range:npt=0-83.660000
a=3GPP-QoE-Metrics:metrics={Initial_Buffering_Duration|Rebuffering_Duration|
   Network_Resource};rate=End;resolution=20
a=control:*
m=video 0 RTP/AVP 96
b=AS:28
```

```
a=3GPP-QoE-Metrics:metrics={Corruption_Duration};rate=End;range:npt=0-40
a=control:trackID=3
a=rtpmap:96 MP4V-ES/1000
a=range:npt=0-83.666000
a=fmtp:96profile-level-id=8;config=000001b008000001b50900012000
m=audio 0 RTP/AVP 98
b=AS:13
a=3GPP-QoE-Metrics:metrics={Corruption_Duration};rate=End;resolution=10
a=control:trackID=5
a=rtpmap:98 AMR/8000
a=range:npt=0-83.660000
a=fmtp:98 octet-align=1
a=maxptime:200
```

8.5 Using MBMS Streaming delivery on Unicast

If the MBMS UE supports MBMS streaming delivery on unicast, then MBMS Streaming shall perform the functions of a PSS client [47] to deliver content when MBMS Bearers are not usable or available and if an alternativeAccessDelivery element is available for the delivery method in the MBMS User Service Description. Note, if an alternativeAccessDelivery element is available, it is presumed that the same content is offered over both PSS and MBMS. If more than one unicastAccessURI element is available in the alternativeAccessDelivery element, then the UE shall randomly choose one URI to be used for unicast access to the service.

If the MBMS UE is receiving an MBMS Streaming User Service using MBMS delivery on Unicast in E-UTRAN, while not being in time-shifting mode, but is interested to receive the corresponding service via an MBMS Bearer instead, it shall handle counting requests as defined in TS 36.300 [96] and TS 36.331 [97].

MBMS and PSS define Quality of Experience (QoE) metrics features. The UE shall not mix MBMS and PSS QoE metrics gatherings and/or reports. QoE is negotiated, gathered and reported separately for PSS and MBMS.

The UE may compare the SSRC values of PSS and MBMS flows. If the UE detects that the same SSRC value is used for PSS and MBMS flows, then the UE should assume that the same wallclock time and same random RTP timestamp offset is used for the flows with the same SSRC value. This gives the UE an advantage for the synchronization onto the flows. The UE does not need to wait for new, flow specific RTCP packets.

The BM-SC and the PSS servers which provide for unicast access to the MBMS service shall be time synchronized. The PSS server and the UE should support UTC clock time format in the “Range” header field as defined in [88].

The UE may request a specific start time of the PSS session by indicating a UTC clock time in the “Range” header field of the PLAY request. The UTC clock time represents the requested streaming start point according to the timeline of the BM-SC. This time may be calculated using the NTP timestamp of the last received RTCP sender reports. Otherwise, the UE may either specify an NPT range using the “now“ value as a start point of the PSS session or it may completely omit the Range header field.

If the PSS server does not support time shifting and the request contains a range indication (other than “now”) then the PSS server shall reply with the actual range that will be played back. Examples of this are an NPT range header field using “now” or the selected start time in UTC clock time.

A *unicastAccessURI* element of an alternativeAccessDelivery element in the *deliveryMethod* element of the User Service Description may contain an RTSP URI or a reference to an PSS SDP file.

9 Associated delivery procedures

9.1 Introduction

Associated delivery procedures describe general procedures, which start before, during or after the MBMS data transmission phase. They provide auxiliary features to MBMS user services in addition, and in association with, MBMS delivery methods and their sessions. Those procedures that shall only be permitted after the MBMS Data transmission phase may also be described as post-delivery procedures.

To enable future backwards compatibility, clause 9 specifies generic and extensible techniques for a potentially wide range of associated delivery procedures.

Clauses 9.3 and 9.4 specify the associated delivery procedures that are initiated only after an MBMS data transmission phase.

The present document describes the following associated delivery procedures:

- File repair, for post-delivery repair of files initially delivered as part of an MBMS download session.
- Content reception reporting of files delivered to an MBMS UE.

These procedures are enabled by establishing a point-to-point connection; and using the MBMS session parameters, received during User Service Discovery/Announcement, to communicate the context (e.g. file and session in question) to the network and the MBMS sender infrastructure. To avoid network congestion in the uplink and downlink directions, and also to protect servers against overload situations, the associated delivery procedures from different MBMS UEs shall be distributed over time and resources (network elements).

One or more *serviceURI* elements in the Associated Delivery Procedure Description specify symbol-based file repair and/or reception report servers. In MBMS download delivery, the use of the "*Alternate-Content-Location-1*" or "*Alternate-Content-Location-2*" elements alone, or in combination with the "Base-URL-1" or "Base-URL-2" elements in the FDT specify standard HTTP/1.1 servers in support of byte-range-based file repair. The network can selectively enable or disable the use of confidentiality protection of Reception Reporting and/or File Repair, based on indicating in the server identities the use of the 'HTTPS' or 'HTTP' scheme as specified in TS 33.246 [10] clause 6.7.

NOTE: The use of the HTTPS scheme for Reception Reporting or File Repair Associated Delivery Procedures is restricted to servers for which a trusted root certificate is present in the list described in TS 33.246 [10], clause 6.7.3.

An instance of an "associated procedure description" is an XML file that describes the configuration parameters of one or more associated delivery procedures.

MBMS Download receivers shall support the file repair procedure as defined in sub-clause 9.3.

MBMS Download receivers shall support the reception reporting procedure as defined in sub-clause 9.4.

MBMS Streaming receivers shall support reception reporting procedures (StaR and StaR-all report types) as defined in sub-clause 9.4.

9.2 Associated Procedure Description

An associated procedure description instance (configuration file) for the associated delivery procedures may be delivered to the MBMS clients:

- during a User Service Discovery / Announcement prior to the MBMS Download delivery session along with the session description (out-of-band of that session); or
- in-band within a MBMS Download delivery session.

The most recently delivered configuration file (i.e. the one with the highest version number - as given from the envelope, see sub-clause 11.1.3) shall take priority, such that configuration parameters received prior to, and out-of-band of, the download session they apply to are regarded as "initial defaults", and configuration parameters received during, and in-band with the download session, overwrite the earlier received parameters. Thus, a method to update parameters dynamically on a short time-scale is provided but, as would be desirable where dynamics are minimal, is not mandatory.

During the User Service Discovery / Announcement Procedure, the associated procedure description instance is clearly identified using a URI, to enable UE cross-referencing of in and out-of-band configuration files.

The MIME application type "application/mbms-associated-procedure-description+xml" as defined in clause C.7 identifies associated delivery procedure description instances (configuration files).

In XML, each associated delivery procedure entry shall be configured using an "`associatedProcedureDescription`" element. All configuration parameters of one associated delivery procedure are contained as attributes of an "`associatedProcedureDescription`" element. The elements (e.g. "`postFileRepair`" and "`postReceptionReport`") of an "`associatedProcedureDescription`" element identify which associated procedure(s) to configure. The associated delivery procedure description is specified formally as an XML schema in sub-clause 9.5.1.

9.3 File Repair Procedure

9.3.1 Introduction

The purpose of the File Repair Procedure is to repair lost or corrupted file fragments from the MBMS download data transmission. When in multicast/broadcast environment, scalability becomes an important issue as the number of MBMS clients grows. Three problems must generally be avoided:

- Feedback implosion due to a large number of MBMS clients requesting simultaneous file repairs. This would congest the uplink network channel.
- Downlink network channel congestion to transport the repair data, as a consequence of the simultaneous clients requests.
- File repair server overload, caused again by the incoming and outgoing traffic due to the clients' requests arriving at the server, and the server responses to serve these repair requests.

The three problems are interrelated and must be addressed at the same time, in order to guarantee a scalable and efficient solution for MBMS file repair.

The principle to protect network resources is to spread the file repair request load in time and across multiple servers.

The MBMS client:

1. Identifies the end of transmission of files or sessions.
2. Identifies the missing data from an MBMS download.
3. Calculates a random *back-off time* and selects a file repair server randomly out of a list.
4. Sends a *repair request* message to the selected file repair server at the calculated time.

When a MBMS download session of repair data is configured in the associated delivery descriptions, a MBMS client should wait for repair data in the defined MBMS download session on its MBMS bearer - except where the UE is prevented from doing so due to limited simultaneous context activation capability.

Then the file repair server:

1. Responds with a *repair response* message either containing the requested data, redirecting the client to an MBMS download session, redirecting the client to another server, or alternatively, describing an error case.

The BM-SC may also send the repair data on a MBMS bearer (possibly the same MBMS bearer as the original download) as a function of the repair process.

The random distribution, in time, of *repair request* messages enhances system scalability to the total number of such messages the system can handle without failure.

9.3.2 Starting Time of the Associated Delivery Procedure for MBMS Download Delivery

FLUTE File Delivery Table (FDT) Instances include an "expires" attribute, which defines the expiration time of the FDT instance. The sender must use an expiry time relative to the current time at the BM-SC. According to RFC 3926 [9], "the receiver SHOULD NOT use a received FDT Instance to interpret packets received beyond the expiration time of the FDT Instance".

The starting time of Associated Delivery Procedure for the MBMS download is the expiration time of the FDT instance at the latest.

The starting time of the *postFileRepair* timer (see sub-clause 9.4.4) corresponds to the starting time of the Associated Delivery Procedure. The *postFileRepair* timer value corresponds to the back-off time determined from sub-clause 9.3.4, using the file repair associated parameters in the ADP.

The starting time for the *postReceptionReport* timer (see clause 9.4.4) for RAck reports corresponds to the time at which the UE determines that there has been a complete file reception for MBMS download, as specified in sub-clause 9.4.1.

The starting time for the *postReceptionReport* timer (see sub-clause 9.4.4) for StaR/StaR-only/StaR-all reports corresponds to the time at which the UE has identified a complete MBMS delivery session reception, as specified in sub-clause 9.4.2.

The *postReceptionReport* timer value for RAck and StaR/StaR-only/StaR-all reports is set to the same value and corresponds to the back-off time, as determined from sub-clause 9.3.4, using the reception reporting associated parameters in the ADP.

The MBMS UE may also choose to start the Associated Delivery Procedure before the FDT instance expires

- when it has received an end-of-object (B-flag) for an object or
- when an end-of-session (A-flag) is received before the FDT instance expires. Note, the end-of session (A-flag) indicates, that neither more objects nor FDT instances will be transmissed by the BM-SC.

If the MBMS UE is not capable of receiveing an MBMS transmission while using an interactive bearer, the MBMS UE shall ignore the end-of-object flags (B-flag).

When a particular file (URI) is present in several FDT Instances with different TOI values, then the FDT Instance with the highest FDT Instance ID defines the TOI for the most recent instance of the file and determines the end of transmission time for that file. A UE shall only determine transmission completeness for a file for the most recent instance of the file - and shall not use FDT Instance expiry time to determine transmission completeness for any other (TOI) instances of a file (fileURI).

NOTE 1: The intention of this sub-clause is to just start the Associated Delivery Procedure back-off timer for the more recent instance version of a file with respect to the FLUTE transmission session.

When a particular file (URI) is present in more than one FDT Instance with the same TOI value, then the end of transmission time is defined by the expiration time of the latest FDT Instance to expire.

If an FDT Instance is received describing the file after this time (giving an FDT Instance expiry time in the future and a different TOI value) the UE shall determine that the transmission of the file is incomplete - i.e. that more packets may arrive within the MBMS download session for that file, 'forgetting' its previous file transmission complete determination.

NOTE 2: This effectively resets and stops any running timers already initiated for an associated delivery procedure for that file.

If the MBMS UE receives an end-of-object packet (with FLUTE header B flag set true) the MBMS UE shall determine that the transmission of that object is complete, and shall interpret that as file transmission complete if no, more recent, TOIs are described for the same file (URI) in any received and unexpired FDT Instance(s).

If the MBMS UE determines that the download session is complete (as specified in sub-clause 9.4.2) then it shall interpret this also that all the transmissions of all files (and TOIs) described by all FDT Instances, received from that session, are complete.

9.3.3 Identification of Missing Data from an MBMS Download

The session description and the MBMS download delivery protocol, FLUTE, provide the client with sufficient information to determine the source block and encoding symbol structure of each file. From this a client is able to determine which source symbols should have been transmitted but have not been received. The client is also able to determine the number of symbols it has received for each source block of each file, and thus the number of further symbols required to decode the block.

Thus, an MBMS client is able to identify any source symbols lost in transmission, and the number (and ESI values where appropriate) of required source and/or repair symbols that would complete the reconstruction of a source block (of a file).

When the MBMS FEC scheme is used, the MBMS client shall consider already received repair symbols when making the determination of the further symbols required. In this case, the client should either:

- identify a minimal set of specific symbols that, combined with the already received symbols, allows the MBMS FEC decoder to recover the file, or
- identify a number, *r*, of symbols such that reception of *r* previously unreceived symbols will allow the MBMS FEC decoder to recover the file.

9.3.4 Back-off Timing the Procedure Initiation Messaging for Scalability

This clause describes a *back-off mode* for MBMS download to provide information on when a receiver, that did not correctly receive some data from the MBMS sender during a transmission session, can start a request for a repair session. In the following it is specified how the information and method a MBMS client uses to calculate a time (*back-off time*), instance of the back-off mode, to send a file repair message to the MBMS server.

The back-off mode is represented by a *back-off unit*, a *back-off value*, and a *back-off window*. The two latter parameters describe the back-off time used by the MBMS client.

The back-off unit (in the time dimension) defaults to *seconds* and it is not signalled.

The *back-off time* shall be given by an *offset time* (describing the back-off value) and a *random time period* (describing the back-off window) as described in the following clauses.

An MBMS client shall generate random or pseudo-random time dispersion of *repair requests* to be sent from the receiver (MBMS client) to the sender (MBMS server). In this way, the repair request is delayed by a pre-determined (random) amount of time.

The back-off timing of *repair request* messages (i.e. delaying the sending of *repair requests* at the receiver) enhances system scalability to the total number of such messages the system can handle without failure.

9.3.4.1 Offset time

The *OffsetTime* refers to the repair request suppression time to wait before requesting repair, or in other words, it is the time that a MBMS client shall wait after the end of the MBMS data transmission to start the file repair procedure. An associated procedure description instance shall specify the wait time (expressed in *back-off unit*) using the "offset-time" attribute.

9.3.4.2 Random Time Period

The *Random Time Period* refers to the time window length over which a MBMS client shall calculate a *random time* for the initiation of the file repair procedure. The method provides for statistically uniform distribution over a relevant period of time. An associated procedure description instance shall specify the wait time (expressed in *back-off unit*) using the "random-time-period" attribute.

The MBMS client shall calculate a uniformly distributed *Random Time* out of the interval between 0 and *Random Time Period*.

9.3.4.3 Back-off Time

The sending of the file *repair request* message shall start at *Back-off Time = offset-time + Random Time*, and this calculated time shall be a relative time after the MBMS data transmission. The MBMS client shall not start sending the repair request message before this calculated time has elapsed after the initial transmission ends.

9.3.4.4 Reset of the Back-off Timer

The reception of an updated (higher version number) associatedDeliveryProcedureDescription and/or an updated sessionDescription shall overwrite the timer parameters used in the back-off algorithm. Except in the case that the offset-time, random-time-period and session end time parameters are identical to the earlier version; the back-off time shall be recalculated. For currently running timers this requires a reset.

9.3.5 File Repair Server Selection

9.3.5.1 List of Server URIs

A list of symbol-based file repair service URIs is provided as elements of the Associated Delivery procedure fragment's *postFileRepair* element. A list of byte-range based repair servers may be additionally provided as elements of the FDT. Service URIs host identity may also be given as IP addresses, which may be used to avoid a requirement for DNS messaging. The file repair service URIs of a single associated delivery procedure description shall be of the same type, e.g. all IP addresses of the same version, or all domain names. The number of symbol-based file repair service URIs is determined by the number of "serviceURI" elements, each of which shall be a child-element of the "procedure" element. The "serviceURI" element provides the references to the file repair server's resource via the "xs:anyURI" value. At least one "serviceURI" element shall be present. The number of byte-range based file repair service URIs is determined by the number of "Alternate-Content-Location-1" and "Alternate-Content-Location-2" elements in the FDT. The "Alternate-Content-Location-1" and "Alternate-Content-Location-2" elements provide the references to the file repair server's resource via the "xs:anyURI" value. At least one "Alternate-Content-Location-1" element shall be present in the FDT if byte-range based file repair is to be supported by the network.

When present, the "Base-URL-1" and "Base-URL-2" elements provide base URLs against which to resolve a relative reference included in any "Alternate-Content-Location-1" or "Alternate-Content-Location-2" element, respectively.

When present, the “Availability-Time” attribute provides a method to inform the UE of an absolute time according to the UTC time standard until which the UE can expect that, if reachable and functioning, the file repair server will return the requested repair data.

9.3.5.2 Selection from the Server URI List

There may be one or more file repair URIs of one or more types present in the Associated Delivery procedure fragment and the FDT. Within a list, the UE randomly selects one of the service URIs from the list, with uniform distribution.

The MBMS client shall exhaust (according to section 9.3.7.1) the list of highest priority URIs before moving to the list of next highest priority, etc.

The priority of file repair URI lists is:

- byte-range based repair servers included as “Alternate-Content-Location-1”
- byte-range based repair servers included as “Alternate-Content-Location-2”
- symbol-based repair servers

9.3.6 File Repair Request Messages

9.3.6.0 General

Once missing file data is identified, the MBMS client sends one or more messages to a file repair server requesting transmission of data that allows recovery of missing file data. All file repair requests and repair responses for a particular MBMS transmission shall take place in a single TCP session using the HTTP protocol (RFC 2616 [18]). The repair request is routed to the file repair server IP address resolved from the selected file repair server URI.

The timing of the opening of the TCP connection to the server, and the first repair request, of a particular MBMS client is randomized over a time window as described in sub-clause 9.3.2. If there is more than one repair request to be made these are sent immediately after the first.

When a MBMS UE identifies symbols or the byte range of symbols in repair requests these symbols shall be source symbols, and should include all the missing source symbols of the relevant source block. Note, these represent information for the file repair server and the BM-SC may use these source symbols and/or redundant symbols in providing the necessary repair data.

After the MBMS download session, the receiver identifies a set of encoding symbols that allow recovery of the missing file data and requests for their transmission in a file repair session.

There are two formats for the MBMS UE to request repair data: the Symbol-Based File Repair Request Message and the Byte-Range-Based Request Messsage.

9.3.6.1 Symbol-Based File Repair Request Message Format

In this message format, the MBMS UE requests specific encoding symbols and uniquely identifies these by the combination (URI, SBN, ESI). This message format shall be used if the MBMS UE is requesting symbols from a file repair server that only supports symbol-based file repair request messages, i.e., the server is listed in a "serviceURI" element of the Associated Delivery procedure. The file repair request shall either include the URI of the file for which it is requesting the repair data or an identifier of a set of files. The URI uniquely identifies the file (resource) and is found from the FLUTE FDT Instances. Additionally, the repair request for single files shall contain the MD5 hash value of the file, if present in the FDT instance declaring the file from which data is being requested. The MD5 hash value is used to identify a specific version of the file.

For completely missed files, a Repair Request may give only the URI of the file and optionally the MD5 hash value of the file. If the MD5 hash value is not present, the server shall respond with the latest version of the file.

A set of files may be fetched using the File Repair server. A client may request all files from a specific FDT instance or a specific logical group of a particular MBMS User Services.

The client makes a file repair request using the HTTP (RFC 2616 [18]) request method GET. Further arguments are encoded into the URI query part (RFC 3986 [19]) as defined below and included in the HTTP GET request. If a number of previously unreceived symbols are requested for a specific Source Block, then the SBN is provided along with the ESI of the symbol, which is subsequent in the symbol sequence to the latest received symbol for that source block and the number of symbols requested. If a number of previously unreceived source blocks are requested for a specific file, the URI should be provided along with an SBN range starting from the first missing source block and ending with the SBN of the last missing source block of the contiguous set of source block. Examples for requesting contiguous and non-contiguous ranges of symbols and source blocks or even entire files or group of files are given below.

For example, assume that in a MBMS download session a 3gp file with URI = www.example.com/news/latest.3gp was delivered to an MBMS client. After the MBMS download session, the MBMS client recognized that it did not receive two packets with SBN = 5, ESI = 12 and SBN=20, ESI = 27. If the selected repair service URI (from the associated delivery procedure meta data fragment) is http://mbmsrepair1.example.com/path/repair_script, only supports symbol-based file repair requests, and the MD5 value of that file is “ODZiYTU1OTFkZGY2NWY5ODh==”, then the HTTP GET request is as follows:

GET /path/repair_script?fileURI=www.example.com/news/latest.3gp&Content-MD5= ODZiYTU1OTFkZGY2NWY5ODh== &SBN=5;ESI=12&SBN=20;ESI=27 **HTTP/1.1**

Host: mbmsrepair1.example.com

A file repair session shall be used to recover the missing file data from a single MBMS download session only. If more than one file were downloaded in a particular MBMS download session, and, if the MBMS client needs repair data for more than one file received in that session, the MBMS client shall send separate HTTP GET requests for each file.

An HTTP client implementation might limit the length of the URL to a finite value, for example 256 bytes. In the case that the length of the URL-encoded (SBN, ESI) data exceeds this limit, the MBMS client shall distribute the URL-encoded data into multiple HTTP GET requests.

In any case, all the HTTP GETs of a single file repair session shall be performed within a single TCP session and they shall be performed immediately one after the other.

In the following, we give the details of the syntax used for the above request method in ABNF.

In this case an HTTP GET with a normal query shall be used to request the missing data, according to HTTP1.1 [RFC2616 [18]]

- repair_request_http_URL = repair_service_URI "?" query
- repair_service_URI = <selected serviceURI from the Associated Delivery Procedure Description>

Where, for MBMS File Repair Request:

- query = std_query | alt_query

- std_query = file_uri ["&" content_md5] *("&" sbn_info)
- file_uri = "fileURI=" URI-reference; URI-reference is as defined in [19].
- content_md5 = "Content-MD5=" 1*(ALPHA / DIGIT / "+" / "/" / "=")
- sbn_info = "SBN=" sbn_range
- sbn_range = (sbnA ["-" sbnZ]) / (sbnA [";" esi_info])
- esi_info = "ESI=" ((esi_range *("," esi_range))) / (esiA “+” number_symbols)
- esi_range = esiA ["-" esiZ]
- sbnA = 1*DIGIT; the SBN, or the first of a range of SBNs
- sbnZ = 1*DIGIT; the last SBN of a range of SBNs
- esiA = 1*DIGIT; the ESI, or the first of a range of ESIs
- esiZ = 1*DIGIT; the last ESI of a range of ESIs
- number_symbols = 1*DIGIT; the number of additional symbols required
- alt_query = service_id “&” (fdt_inst_id | fdt_group_id)
- service_id = "serviceId=" <value of the *serviceId* attribute of the User Service Description>
- fdt_inst_id = “fdtInstanceId=” <as defined in clause 3.4.1 of [9] or in clause 7.3>
- fdt_group_id = “fdtGroupId=” <as defined in clause 7.2.9>

Thus, the following symbols adopt a special meaning for MBMS download URI: ? - + , ; & =

One example of a query on encoding symbol 34 of source block 12 of a music file “www.example.comm/greatmusic/number1.aac" using the provided repair service URI "http://mbmsrepair1.example.com/path/repair_script" is:

- http://mbmsrepair1.example.com/path/repair_script?fileURI= www.example.com/greatmusic/number1.aac&SBN=12;ESI=34

An example of requesting an entire file is

- http://mbmsrepair1.example.com/path/repair_script?fileURI= www.example.com/greatmusic/number1.aac

An example of requesting a specific source block from a specific file version is

- http://mbmsrepair1.example.com/path/repair_script?fileURI= www.example.com/greatmusic/number1.aac&Content-MD5=ODZiYTU1OTFkZGY2NWY5ODh==

For messaging efficiency, the formal definition enables several contiguous and non-contiguous ranges to be expressed, as well as a number of symbols with ESIs of a given value or above in a single query:

- An entire file (like in the above example).
- A symbol of a source block (e.g. ...&SBN=12;ESI=23).
- A range of symbols for a certain source block (e.g. ...&SBN=12;ESI=23-28).
- A number of symbols with ESIs of a given value or above (e.g. …&SBN=12;ESI=120+10).
- A list of symbols for a certain source block (e.g. ...&SBN=12;ESI=23,26,28).
- All symbols of a source block (e.g. ...&SBN=12).
- All symbols of a range of source blocks (e.g. ...&SBN=12-19).

- non-contiguous ranges (e.g.1. ...&SBN=12;ESI=34&SBN=20;ESI=23 also, e.g. 2. ...&SBN=12-19&SBN=28;ESI=23-59&SBN=30;ESI=101).

An example to request all file of a particular FDT instance is given below:

- http://mbmsrepair1.example.com/path/repair_script?serviceId=urn:3gpp:0010120123hotdog&fdtInstanceId=12

9.3.6.2 Byte-Range-Based File Repair Request Message Format

In this message format, the MBMS UE uses the conventional HTTP/1.1 GET or partial GET requests as defined in RFC 2616 [18] to request all or a subset of source symbols of the referenced resource, respectively. The UE shall support these message requests formats to allow the file repair requests to be serviced by a standard HTTP/1.1 server. These message formats shall be used if the MBMS UE is requesting symbols from a file repair server that supports byte range requests, i.e, the server is listed in the "Alternate-Content-Location-1" or "Alternate-Content-Location-2" elements in the FDT.

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

If the the MBMS UE only requests transmission of a subset of the source symbols or sub-symbols the UE uses the HTTP partial GET request with the `Range` request header as defined in 14.35.2 of RFC 2616 [18]. The MBMS UE shall indicate the specific source symbols or sub-symbols as a `byte-range-spec` as defined in 14.35.1 of RFC 2616 [18].

For messaging efficiency, the HTTP GET method allows the UE to include multiple byte range requests within a single partial GET request. If the UE 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 MBMS UE determines that it can select among multiple subsets of the source symbols or sub-symbols, the MBMS UE 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 file repair servers.

If more than one file were downloaded in a particular MBMS download session, and, if the MBMS client needs repair data for more than one file received in that session, the MBMS client shall send separate HTTP GET requests for each file.

If the "Content-MD5" element is included in the FDT Instance for the file, then the MBMS UE shall use the "If-Range" or the "If-Match" HTTP protocol headers together with the "Range" header in the following way. The "Content-MD5" value that is given in the FDT shall be used as the entity tag. The UE should use the "If-Match" header to receive the requested range(s) of the file associated with the entity-tag, or no repair data if the request cannot be satisfied by the repair server. The UE may use the "If-Range" header if it wishes to receive the latest version of the entire file in case the version associated with the entity-tag is no longer available on the repair server. To reduce the impact to capacity, the UE should not use the "If-Range" header if it can request the range(s) from other repair servers.

For example, assume that the "Alternate-Content-Location-1" element in the FDT of the file indicates that byte range repair requests are supported by the HTTP server at URI www.example.com/service1/news_service/latest_news.mp4. The UE determines that it requires the byte ranges 5018640-5042399 and 19037040-19050239. The MD5 provided in the FDT of the file is Base64 encoded as B2B359591E961C6B0F468FE536BCD920. The HTTP GET request may look as follows:

GET /service1/news_service /latest_news.mp4 **HTTP/1.1**
If-Match: "B2B359591E961C6B0F468FE536BCD920="
Range: bytes=5018640-5042399,19037040-19050239
Host: www.example.com

In case the version identifier, indicated as the MD5 value 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

9.3.7 File Repair Response Message

Once the MBMS file repair server has assembled a set of encoding symbols that contain sufficient data to allow the UE to reconstruct the file data from a particular file repair request, the MBMS file repair server sends one message to the UE. Each file repair response occurs in the same TCP and HTTP session as the repair request that initiated it.

An MBMS client shall be prepared for any of these 5 response scenarios:

- The server returns a repair response message where a set of encoding symbols forms an HTTP payload as specified below (see 9.3.7.2 for details).
- The server returns a repair response message where a byte range or set of byte ranges forms an HTTP payload as specified below (see 9.3.7.2a for details).
- The server returns the requested file or file groups (see 9.3.7.5 for details).
- The server redirects the client to a broadcast/multicast delivery (an MBMS download session).
- The server redirects the client to another file repair server (if a server is functioning correctly but is temporarily overloaded).
- An HTTP error code is returned (note that sub-clause 9.3.8 describes the case of no server response).

For (reasonably) uniformly distributed random data losses, immediate point-to-point HTTP delivery of the repair data will generally be suitable for all clients. However, broadcast/multicast delivery of the requested data may be desirable in some cases:

- A repeat MBMS download (all or part of the files from a download session) is already scheduled and the BM-SC prefers to handle repairs after that repeat MBMS download.
- Many UEs request download data (over a short period of time) indicating that broadcast/multicast delivery of the repaired data would be desirable.

In this case a redirect to the broadcast/multicast repair session for UEs that have made a repair request would be advantageous.

9.3.7.1 Symbol-Based File Repair Response Messages Codes

The response codes of HTTP servers to the byte-range-based repair request message in 9.3.6.2 are specified in RFC 2616 [18]. The response codes of symbol-based file repair servers to the symbol-based repair request message in 9.3.6.1 are specified as follows.

In the case that the file repair server receives a correctly formatted repair request which it is able to understand and properly respond to with the appropriate repair data, the file repair server shall attempt to serve that request without an error case.

For a direct point-to-point HTTP response with the requested data, the file response message shall report a 200 OK status code and the file repair response message shall consist of HTTP header and file repair response payload (HTTP payload), as defined in sub-clause 9.3.7.2. If the client receives a 200 OK response with fewer than all the quantity of requested symbols it shall assume that the file repair server wishes the missing symbols to be requested again (due to its choice or inability to deliver those symbols with this HTTP response).

For a redirect case the file repair server uses the HTTP response status code 302 (Found - Redirection) to indicate to the UE that the resource (file repair data) is temporarily available via a different URI. The temporary URI is given by the Location field in the HTTP response. In the case of a redirect to another file repair server, this temporary URI shall be the URL of that repair server.

In the case of a redirect to a broadcast/multicast delivery, the temporary URI shall be the URI of the Session Description (SDP file) of the broadcast/multicast (repair) session as described in sub-clause 9.3.7.3.Other HTTP status

codes (RFC 2616 [18]) shall be used to support other cases. Other cases may include server errors, client errors (in the file repair request message) and server overload.

In case the file repair server does not find the requested file (file with given fileURI is found), the server shall respond with “400 Bad Request” and optionally with “0001 File not found” in the response body. As a result, the MBMS UE may choose another file repair server as defined in clause 9.3.5.

In case the file repair server does not find the requested version of the requested file (file with given fileURI is found but Content-MD5 is not found), the server shall respond with “400 Bad Request” and optionally with “0002 Content-MD5 not valid” in the response body. As a result, the MBMS UE may chose another file repair server as defined in clause 9.3.5. Or the MBMS UE may request the latest version of the file and discard the previously received chunks of the file. Note, the MBMS UE can request the latest version of a file by using only the fileURI argument in the file repair request.

Note. In case of repetitive server errors, the client is not expected to go through the complete list of available file repair servers, and may abandon after a limited number of attempts.

In case the file repair server does not find any of the requested SBN or ESI values, it shall respond with the “400 Bad Request” and optionally with “0003 SBN or ESI out of range” in the respon se body. As a result, the UE should discard all received chunks of the file and request the entire file from the file repair server.

In case the file repair server receives unknown query line arguments, it shall respond with “501 Not Implemented”. The server should add the HTTP1.1 “Server” header with the value “MBMS/6”. As a result, the client should try to fetch the entire file from the file repair server. Note, this behaviour is intended to make the file repair service forward compatible and allow addition of new function in later releases.

In case the file repair server does not find the requested serviceId value, it shall respond with the “400 Bad Request” and optionally with “0004 ServiceId not found” in the response body. As a result, the UE should request the needed file separately using the fileURI query line argument.

In case the file repair server does not find the requested fdtInstanceId value, it shall respond with the “400 Bad Request” and optionally with “0005 fdtInstanceId not found” in the response body. As a result, the UE should request the needed file separately using the fileURI query line argument.

In case the file repair server does not find the requested fdtGroupId value, it shall respond with the “400 Bad Request” and optionally with “0006 fdtGroupId not found” in the response body. As a result, the UE should request the needed file separately using the fileURI query line argument.

In case the file repair server is,or is about to, experiencing an overload condition, it should respond with the “503 Service Unavailable” that can include a *Retry-After* header. As a result, the UE should stop the file repair procedure to that file repair server. The UE shall consider this server unavailable for this file repair session, or, if supported by the UE, for the period of time indicated in the *Retry-After* header, The UE may immediately try an alternative available file repair server. The UE may re-try the current file repair server after the *Retry-After* time has elapsed. In the case that all known file repair servers have been exhausted in this manner, the UE shall cease the file repair procedure. When the time in Retry-After header is expressed as an integer number of seconds then it is relative to the reception time of the “503 Service Unavailable”.

HTTP response error messages may contain a message body, which gives a more detailed error message. The MIME type of such message body shall be in text/plain. The syntax of the HTTP error message body is defined in ABNF [23] as follows:

```
http-error-body = error-code (SP / HTAB) error-description CRLF

error-code = 4DIGIT

error-description = 1*(SP / VCHAR)
```

Note that the following error messages MAY be used in the message body of the HTTP response error messages.

0001 File not found

0002 Content-MD5 not valid

0003 SBN or ESI out of range

0004 ServiceId not found

0005 fdtInstanceId not found

0006 fdtGroupId not found

9.3.7.2 Symbol-Based File Repair Response Message Format for HTTP Carriage of Repair Data

The format of the response message to the symbol-based repair request message in 9.3.6.1 is specified here.

The file repair response message consists of HTTP header and file repair response payload (HTTP payload).

The HTTP header shall provide:

- HTTP status code, set to 200 OK for the case of a successful request.
- Content type of the HTTP payload (see below).

NOTE: Other HTTP headers (RFC 2616 [18]) may also be used but are not mandated by this mechanism.

The Content-Type shall be set to "application/simpleSymbolContainer", which denotes that the message body is a simple container of encoding symbols as described below.

This header is as follows:

- HTTP/1.1 200 OK
- Content-Type: application/simpleSymbolContainer

NOTE: Other HTTP headers (RFC 2616 [18]) may also be used but are not mandated by this mechanism.

Encoding symbols are included in the response in groups. Each group is preceded by an indication of the number of symbols within the group and an FEC Payload ID coded according to the FEC scheme used for the original file delivery session. The FEC Payload ID identifies all the symbols in the group in the same way that the FEC Payload ID of an FEC source or repair packet identifies all the symbols in the packet. The file repair response payload is constructed by including each FEC Payload ID and Encoding Symbol group one after another (these are already byte aligned). The order of these pairs in the repair response payload may be in order of increasing SBN, and then increasing ESI, value; however no particular order is mandated.

A single HTTP repair response message shall contain, at the most, the same number of symbols as requested by the respective HTTP repair request message.

The UE and file repair server already have sufficient information to calculate the length of each encoding symbol and each FEC Payload ID. All encoding symbols are the same length; with the possible exception of the last source encoding symbol in the repair response. All FEC Payload IDs are the same length for one file repair request-response as a single FEC Scheme is used for a single file.

Figure 17: deleted

Figure 18 illustrates the complete file repair response message format (box sizes are not indicative of the relative lengths of the labelled entities).

HTTP Header		
Length Indicator	FEC Payload ID	Encoding Symbols
Length Indicator	FEC Payload ID	Encoding Symbols
Length Indicator	FEC Payload ID	Encoding Symbols

Length Indicator (2 bytes): indicates the number of encoding symbols in the group (in network byte order, i.e. high order byte first)
FEC Payload ID: indicates which encoding symbols are included in the group. The format and interpretation of the FEC Payload ID are dependent on the FEC Scheme in use.
Encoding Symbols: contain the encoding symbols. All the symbols shall be the same length.

Figure 18: File Repair Response Message Format

9.3.7.2a Byte-Range-Based File Repair Response Message Format for HTTP Carriage of Repair Data

The response message to the byte-range-based repair request message in 9.3.6.2 follows the format and procedures in RFC 2616 [18] 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 [18].

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 [18].

9.3.7.3 File Repair Response for Broadcast/Multicast of Repair Data

Details of how a file repair server decides, or is instructed, to use broadcast/multicast repair instead of point-to-point over HTTP are implementation specific and beyond the scope of the present document.

Prior to the decision to use broadcast/multicast repair, each repair response shall be provided by HTTP according to sub-clause 9.3.7.2 or 9.3.7.2a.

The file repair server uses the HTTP response status code 302 (Found - Redirection) to indicate to the UE that the resource (file repair data) is temporarily available via a different URI. The temporary URI is given by the Location field in the HTTP response and is the URI of the Session Description (SDP file) of the broadcast/multicast repair session.

Where feasible, it is recommended that the same download session that delivered the original data be used for the broadcast/multicast repair. If this conflicts with the session end time limit of the Session Description then a new version of the Session Description shall be sent with an updated (extended) session end time. This shall be sent in-band of that download session.

In some cases this may not be feasible and a different (possibly new) download session may be defined for the repair.

The SDP file for broadcast/multicast repair session may be carried as payload (entity-body) in the HTTP response - which is especially useful if the broadcast/multicast repair session is a new (or recently end time modified) FLUTE download session and other means of service announcement prior to this were not feasible.

The delivery method's associatedDeliveryProcedureDescription may be updated and the new version transmitted in-band with the download session so that currently active client back-off timers are reset, thus minimizing additional client requests until after the broadcast/multicast repair session. The server shall be prepared for additional requests in any case as successful reception of the updated associatedDeliveryProcedureDescription can not be assured in all cases.

The existence of a broadcast/multicast file repair session is signalled by the inclusion of the optional *bmFileRepair* procedure in the updated Associated Delivery procedure description. This is signalled by the *bmFileRepair* element with a single "sessionDescriptionURI" attribute of type "xs:anyURI" which specifies the URI of the broadcast/multicast file repair session's session description.

In the cases where the same IP addressing is used for the broadcast/multicast repair session as the original download session, the UE simply shall not leave the group. Otherwise, the UE shall join to the MBMS bearer for the repair session as it would for any MBMS session.

A broadcast/multicast file repair session behaves just as an MBMS download session, and the determination of end of files and session, and use of further associated delivery procedures uses the same techniques as specified for the MBMS download delivery method.

9.3.7.4 File Repair Response Message Format for HTTP carriage of Complete Files

The file repair response message consists of HTTP header and one or more complete files.

The HTTP header shall provide:

- HTTP status code, set to 200 OK for the case of a successful response.
- Content type shall be set to multipart/related

NOTE: Other HTTP headers (RFC 2616 [18]) may also be used but are not mandated by this mechanism.

The server shall encapsulate the requested files into a multipart mime container. Each part of the multipart mime shall contain at least the Content-Location of the embedded files.

9.3.8 Server Not Responding Error Case

In the error case where a UE determines that the its selected file repair server is not responding it shall return to the serverURI list of repair servers and uniformly randomly select another server from the list, excluding any servers it has determined are not responding. All the repair requests message(s) from that UE shall then be immediately sent to the newly selected file repair server.

If all of the repair servers from the serverURI list are determined to be not responding, the UE may attempt an HTTP GET to retrieve a, potentially new, instance of the session's Associated Procedure Description; otherwise UE behaviour in this case is unspecified.

A UE determines that a file repair server is not responding if any of these conditions apply:

1. The UE is unable to establish a TCP connection to the server.
2. The server does not respond to any of the HTTP repair requests that have been sent by the UE (it is possible that second and subsequent repair requests are sent before the first repair request is determined to be not-responded-to).
3. The server returns an unrecognized message (not a recognizable HTTP response).
4. The server returns an HTTP server error status code (in the range 500 to 505).

9.4 The Reception Reporting Procedure

Following successful reception of content whether through point-to-multipoint MBMS bearers only, unicast bearers only, or using both point-to-multipoint and point-to-point bearers, a reception reporting procedure can be initiated by the MBMS Receiver (UE) to the BM-SC.

For MBMS Download Delivery method, the reception reporting procedure is used to either report the complete reception of one or more files, or to report statistics on the stream, or to do both. For MBMS Streaming Delivery method, the reception reporting procedure is used to report statistics on the stream.

If the BM-SC provided parameters requiring reception reporting confirmation then the MBMS Receiver shall confirm the content reception.

If reception reporting is requested for statistical purposes the BM-SC may specify the percentage subset of MBMS receivers it would like to perform reception reporting.

Transport errors can prevent an MBMS Receiver from deterministically discovering whether the reception reporting associated delivery procedure is described for a session, and even if this is successful whether a sample percentage is described. An MBMS Receiver shall behave according to the information it has even when it is aware that this may be incomplete.

The MBMS Receiver:

1. Identifies the completion of the reception of an MBMS session and its content items (e.g. a file, or a set of files within a download delivery session). See sub-clauses 9.4.1and 9.4.2.
2. Determines the need to report reception. See sub-clause 9.4.3.
3. Determines the ability to aggregate reception reports. See sub-clause 9.4.4.
4. Selects a time (Request time) at which a reception report request will be sent and selects a server from a list - both randomly and uniformly distributed. See sub-clauses 9.4.4 and 9.4.5.
5. Sends a *reception report request* message to the selected server at the selected time. See sub-clause 9.4.6.

Then the server:

1. Responds with a *reception report response* message either describing a success or an error case. See sub-clause 9.4.7.

9.4.1 Identifying Complete File Reception from MBMS Download and Determining Download Status

A file is determined to be completely downloaded when it is fully received and reconstructed by MBMS reception with FEC decoding (if FEC is actually used) and/or a subsequent File Repair Procedure (sub-clause 9.3).

When compiling RAck reception reports for a download delivery session, the failure or success of a file is determined after FEC decoding and any subsequent File Repair Procedure.

When compiling StaR, StaR-all, or StaR-only reception reports for a download delivery session, the failure or success of a file is determined after FEC decoding and before any subsequent File Repair Procedure.

9.4.2 Identifying Complete MBMS Delivery Session Reception

Delivery sessions (download and streaming) are considered complete when the "time to" value of the session description (from "t=" in SDP) is reached. Where the end time is unbounded (time to = 0) then this parameter is not used for identifying completed sessions.

Delivery sessions are also considered complete when the UE decides to exit the session - where no further data from that session will be received. In this case the UE may or may not deactivate the MBMS bearer(s).

For MBMS download sessions, FLUTE provides a "Close session flag" (see sub-clause 7.2.7) which, when used, indicates to the UE that the session is complete.

9.4.3 Determining Whether a Reception Report Is Required

Upon full reception of a content item or when a session is complete, the MBMS Receiver must determine whether a reception report is required. An Associated Delivery Procedure Description indicates the parameters of a reception reporting procedure (which is transported using the same methods as the ones that describe File Repair).

A delivery method may associate zero or one associated delivery procedure descriptions with an MBMS delivery session. Where an associated delivery procedure description is associated with a session, and the description includes a *postReceptionReport* element, the UE shall initiate a reception reporting procedure. Reception reporting behaviour depends on the parameters given in the description as explained below.

The Reception Reporting Procedure is initiated if:

a. A *postReceptionReport* element is present in the associated procedure description instance, and/or if the QoE reporting according to sub-clause 8.3.2.2 is activated.

In the event that the UE determines a selection for reporting to both active reporting procedures, it should comply with both reporting configurations. Should such separate reporting not be desirable due to, for example, report compilation complexity, reception reporting in accordance to OMA-DM should take precedence over reporting in accordance with the Associated Delivery Procedure.

One of the following will determine the UE behaviour:

b. *reportType* is set to RAck (Reception Acknowledgement). Only successful file reception is reported without reception details.

c. *reportType* is set to StaR (Statistical Reporting for successful reception). Successful file reception is reported (as with RAck) with reception details for statistical analysis in the network.

d. *reportType* is set to StaR-all (Statistical Reporting for all content reception). The same as StaR with the addition that failed reception is also reported. StaR-all is relevant to both streaming and download delivery.

e. *reportType* is set to StaR-only (Statistical Reporting without Reception Acknowledgement). The same as StaR-all with the exception that individual files are not acknowledged. Only reception details are reported for the session for both streaming and download delivery. StaR-only is equivalent to StaR-all for streaming delivery. StaR-all is relevant to download delivery where session performance is obtained through QoE metrics.

The *reportType* attribute is optional and behaviour shall default to RAck when it is not present.

The *samplePercentage* attribute can be used to set a percentage sample of receivers which should report reception. This can be useful for statistical data analysis of large populations while increasing scalability due to reduced total uplink signalling. The *samplePercentage* takes on a value between 0 and 100, including the use of decimals. It is recommended that no more than 3 digits follow a decimal point (e.g. 67.323 is sufficient precision).

The *samplePercentage* attribute is optional and behaviour shall default to 100 (%) when it is not present. The *samplePercentage* attribute may be used with StaR, StaR-only and StaR-all, but shall not be used with RAck.

When the *samplePercentage* is not present or its value is 100 each UE which entered the associated session shall send a reception report. If the *samplePercentage* were provided for reportType StaR, StaR-only and StaR-all and the value is less than 100, the UE generates a random number which is uniformly distributed in the range of 0 to100. The UE sends the reception report when the generated random number is of a lower value than the *samplePercentage* value.

9.4.4 Request Time Selection

The MBMS receiver selects a time at which it is to issue a delivery confirmation request. The default start time for the reception reporting procedure is defined in clause 9.3.2.

Back-off timing is used to spread the load of delivery confirmation requests and responses over time.

Back-off timing is performed according to the procedure described in sub-clause 9.3.4. The *offsetTime* and *randomTimePeriod* used for delivery confirmation may have different values from those used for file-repair and are signalled separately in the reception reporting description of the associated delivery procedure description instance.

In general, reception reporting procedures may be less time critical than file repair procedures. Thus, if a postFileRepair timer may expire earlier than a postReceptionReport, radio and signalling resources may be saved by using the file repair point-to-point PDP context (and radio bearer) activate period also for reception reporting (to remove the delay and signalling of multiple activations and deactivations over time)

The default behaviour is that a UE shall stop its postReceptionReport timers which are active when a postFileRepair timer expires, and the UE shall send the corresponding reception report(s) immediately following the file repair procedure on the point-to-point communication setup for file repair.

In some circumstances, the system bottleneck may be in the server handling of reception reporting. In this case the *forceTimeIndependence* attribute may be used and set to true. (false is the default case and would be a redundant use of this optional attribute). When *forceTimeIndependence* is true the UE shall not use file repair point-to-point connections to send reception reporting messages. Instead it will allow the timers to expire and initiate point-to-point connections dedicated to reception report messaging.

A UE may aggregate multiple reception reports together to be more efficient. When a UE has determined that a reception report is required, the UE may determine that if a currently pending reception report exists, these reception reports may be aggregated and sent together.

In the case that a UE decides to aggregate more than one reception reports together from across different user services, the UE shall collect the new reception report and the postReceptionReport timer for this aggregated bundle shall be set to the lowest remaining postReceptionReport timer of the individual reception reports in the aggregated bundle.

In the case that a UE decides to aggregate more than one reception reports together within a user service, the UE shall collect the new reception report and the postReceptionReport timer for this aggregated bundle shall be set according to the first reception report and remain unchanged during any further aggregation (i.e., equal to the sum of the offsetTime and the generated randomTimePeriod of the first reception report in the aggregated bundle.

In the case of aggregating reception reports from within a user service, if a UE has generated a random number to compare against *samplePercentage* (in the case that *samplePercentage* is used), this number shall persist until the aggregated reception report bundle is sent. This means that an original decision to report or not persists until the end of a session. In the case of aggregating across services, the various generated random numbers to compare against *samplePercentage* remain independent from other services.

These reception reports are then aggregated as in sub-clause 9.4.6 and the aggregated bundle shall be sent at the expiration of the postReceptionReport timer, assuming no further aggregation.

For StaR, StaR-only and StaR-all, session completeness - according to sub-clause 9.4.2 - shall determine the back-off timer initialization time.

For RAck, the complete download session - according to sub-clause 9.4.2 - as well as completing any associated file repair delivery procedure shall determine the back-off timer initialization time. RAcks shall be only sent for completely received files.

9.4.5 Reception Report Server Selection

Reception report server selection is performed according to the procedure described in sub-clause 9.3.5.2.

In the case of an aggregated reception report, the server URI list shall be the subset of server URIs common to all reports.

9.4.6 Reception Report Message

Once the need for reception reporting has been established, the MBMS receiver sends one or more Reception Report messages to the reception report server URI. All Reception Report requests and responses for a particular MBMS transmission should take place in a single TCP session using the HTTP protocol (RFC 2616 [18]).

The Reception Report request shall include the URI of the file for which delivery is being confirmed. URI is required to uniquely identify the file (resource).

The client shall make a Reception Report request using the HTTP (RFC 2616 [18]) POST request carrying XML formatted metadata for each reported received content (file). An HTTP session shall be used to confirm the successful

delivery of a single file. If more than one file were downloaded in a particular MBMS download multiple reception reports shall be added in a single POST request.

Each Reception Report is formatted in XML according the following XML schema (sub-clause 9.5.3). An informative example of a single reception report XML object is also given (sub-clause 9.5.3.2).

Multipart MIME (multipart/mixed) may be used to aggregate several small XML files of reception reports to a larger object.

For Reception Acknowledgement (RAck) a receptionAcknowledgement element shall provide the relevant data.

For Statistical Reporting (StaR) one or more statisticalReport elements shall provide the relevant data.

Multiple reception reports can be aggregated together in order to reduce radio resources and HTTP transactions. In the case that sessions are close together as defined in sub-clause 9.4.4, two or more reception reports should be aggregated together at the client. There are two possible mechanisms for aggregating reception reports. In the case of StaR (StaR, StaR-all, StaR-only):

- a single reception report should contain multiple *statisticalReport* elements, each relating to a different *serviceID*.
- alternatively, multipart MIME (multipart/mixed) may be used to aggregate several reception report XML files.

In the case of RAck reporting:

- multipart MIME (multipart/mixed) may be used to aggregate several reception report XML files.

For both RAck and StaR/StaR-all (mandatory):

- For download, one or more *fileURI* elements shall specify the list of files which are reported. If the Content-MD5 value of the file is present in the FDT, it shall be provided in the Content-MD5 attribute in the reception report. Note, this allows unambiguous identification of the files. For RAck reporting, a *clientID* attribute, whose format is the same as that defined below for *clientId* under StaR/StaR-all/StaR-only, may be included. If present, it shall be included in at least the first instance the *fileURI* elements to allow identification of the client that has received the file(s) identified by all instances of the *fileURI* element in the RAck report.
- For the StaR-all mode only, a list of the number of received symbols and a list of the total number of source symbols shall be provided for failed blocks of the file, if any. Both lists are tabulated before any unicast file repair procedures. Thus, the lists are provided for failed files, and for successfully received files that required unicast file repair procedures.

For only StaR/StaR-all/StaR-only (all optional):

- Each *fileURI* element has an optional *receptionSuccess* status code attribute which defaults to "true" ("1") when not used. This attribute shall be used for StaR-all reports. This attribute shall not be used for StaR reports. This attribute is not relevant for StaR-only reports.
- Each QoE Metrics element has a set of attributes and any number of media level QoE Metrics elements. All attributes are defined in sub-clause 9.5.3 and correspond to the QoE metrics listed in sub-clause 8.4.2. Individual metrics, both at session and at media level can be selected via SDP as described in sub-clause 8.3.2.1.
- The *sessionID* attribute identifies the delivery session. If the *sessionType* is "download", *sessionID* is of the format source_IP_address + ":" + FLUTE_TSI. If the *sessionType* is "streaming", *sessionID* is of the format source_IP_address + ":" + RTP_destination_port.
- The *sessionStartTime* and *sessionStopTime* attributes identifies the NTP time when the session was started and stopped.
- The *sessionType* attribute defines the basic delivery method session type used = "download" || "streaming".
- The *serviceId* attribute is value and format is taken from the respective userServiceDescription serviceID definition.
- The *clientId* attribute is unique identifier for the receiver, e.g. an MSISDN of the UE as defined in [77].

- The *serviceURI* attribute value and format is taken from the respective associatedDeliveryProcedureDescription serviceURI, which was selected by the UE for the current report. This attribute expresses the reception report server to which the reception report is addressed.

9.4.7 Reception Report Response Message

An HTTP response is used as the Reception Report response message.

The HTTP header shall use a status code of 200 OK to signal successful processing of a Reception Report. Other status codes may be used in error cases as defined in RFC 2616 [18].

9.5 XML-Schema for Associated Delivery Procedures

9.5.1 Generic Associated Delivery Procedure Description

Below is the formal XML syntax of associated delivery procedure description instances. Documents following this schema can be identifed with the MIME type "application/mbms-associated-procedure-description+xml" defined in Annex C.7. The schema filename of delivery procedure description is associatedprocedure.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3gpp:metadata:2005:MBMS:associatedProcedure"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3gpp:metadata:2005:MBMS:associatedProcedure"
    elementFormDefault="qualified">
    <xs:element name="associatedProcedureDescription" type="associatedProcedureType"/>
    <xs:complexType name="associatedProcedureType">
        <xs:sequence>
            <xs:element name="postFileRepair" type="basicProcedureType" minOccurs="0"/>
            <xs:element name="bmFileRepair" type="bmFileRepairType" minOccurs="0"/>
            <xs:element name="postReceptionReport" type="reportProcedureType" minOccurs="0"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="basicProcedureType">
        <xs:sequence>
            <xs:element name="serviceURI" type="xs:anyURI" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="offsetTime" type="xs:unsignedLong" use="optional"/>
        <xs:attribute name="randomTimePeriod" type="xs:unsignedLong" use="required"/>
    </xs:complexType>
    <xs:complexType name="bmFileRepairType">
        <xs:attribute name="sessionDescriptionURI" type="xs:anyURI" use="required"/>
    </xs:complexType>
    <xs:complexType name="reportProcedureType">
        <xs:complexContent>
            <xs:extension base="basicProcedureType">
                <xs:attribute name="samplePercentage" type="xs:decimal" use="optional"
                    default="100"/>
                <xs:attribute name="forceTimeIndependence" type="xs:boolean" use="optional"
                    default="false"/>
                <xs:attribute name="reportType" type="xs:string" use="optional"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
</xs:schema>

"reportType" value = "RAck" || "StaR" || "StaR-all" || "StaR-only"
```

9.5.2 Example Associated Delivery Procedure Description Instance

Below is an example of an associated delivery procedure description for reception reporting.

```
<?xml version="1.0" encoding="UTF-8"?>
<associatedProcedureDescription
    xmlns="urn:3gpp:metadata:2005:MBMS:associatedProcedure"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:associatedProcedure associatedprocedure.xsd">
    <postFileRepair
            offsetTime="5"
```

```
            randomTimePeriod="10">
        <serviceURI>http://mbmsrepair0.example.com/path/repair_script</serviceURI>
        <serviceURI>http://mbmsrepair1.example.com/path1/repair_script</serviceURI>
        <serviceURI>http://mbmsrepair2.example.com/path2/repair_script</serviceURI>
    </postFileRepair>
    <bmFileRepair sessionDescriptionURI="http://www.example.com/3gpp/mbms/session1.sdp"/>
    <postReceptionReport
            offsetTime="5"
            randomTimePeriod="10"
            reportType="star-all"
            samplePercentage="100"
            forceTimeIndependence="0">
        <serviceURI>http://mbmsreport.example.com/path/report_script</serviceURI>
    </postReceptionReport>
</associatedProcedureDescription>
```

9.5.3 XML Syntax for a Reception Report Request

Below is the formal XML syntax of reception report request instances. The schema filename of reception report request is receptionreport.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3gpp:metadata:2008:MBMS:receptionreport"
xmlns="urn:3gpp:metadata:2008:MBMS:receptionreport"
    elementFormDefault="qualified">
    <xs:element name="receptionReport" type="receptionReportType"/>
    <xs:complexType name="receptionReportType">
        <xs:choice>
            <xs:element name="receptionAcknowledgement" type="rackType"/>
            <xs:element name="statisticalReport" type="starType"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
        </xs:choice>
    </xs:complexType>

<xs:complexType name="rackType">
    <xs:sequence>
        <xs:element name="fileURI" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="fileUriType">
                        <xs:attribute name="clientId" type="xs:string" use="optional"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="starType">
        <xs:sequence>
            <xs:element name="fileURI" type="fileUriType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="qoeMetrics" type="qoeMetricsType" minOccurs="0" maxOccurs="1"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="sessionType" type="sessionTypeType" use="optional"/>
        <xs:attribute name="serviceId" type="xs:string" use="optional"/>
        <xs:attribute name="clientId" type="xs:string" use="optional"/>
        <xs:attribute name="serviceURI" type="xs:anyURI" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
</xs:complexType>
<xs:simpleType name="sessionTypeType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="download"/>
        <xs:enumeration value="streaming"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="fileUriType">
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:attribute name="receptionSuccess" type="xs:boolean" use="optional" default="true"/>
            <xs:attribute name="Content-MD5" type="xs:base64Binary" use="optional"/>
            <xs:attribute name="receivedSymbolsForFailedBlocks" type="unsignedLongVectorType"
use="optional"/>
            <xs:attribute name="totalSymbolsForFailedBlocks" type="unsignedLongVectorType"
use="optional"/>
```

```
            <xs:anyAttribute processContents="skip"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="qoeMetricsType">
        <xs:sequence>
            <xs:element name="medialevel_qoeMetrics" type="medialevel_qoeMetricsType"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="totalRebufferingDuration" type="doubleVectorType" use="optional"/>
        <xs:attribute name="numberOfRebufferingEvents" type="unsignedLongVectorType"
            use="optional"/>
        <xs:attribute name="initialBufferingDuration" type="xs:double" use="optional"/>
        <xs:attribute name="contentAccessTime" type="xs:double" use="optional"/>
        <xs:attribute name="sessionStartTime" type="xs:unsignedLong"/>
        <xs:attribute name="sessionStopTime" type="xs:unsignedLong"/>
        <xs:attribute name="networkResourceCellId" type="stringVectorType" use="optional"/>
        <xs:attribute name="numberOfLostObjects" type="unsignedLongVectorType"
            use="optional"/>
        <xs:attribute name="symbolCountUnderrun" type="stringVectorType" use="optional"/>
        <xs:attribute name="numberOfReceivedObjects" type="unsignedLongVectorType"
            use="optional"/>
        <xs:anyAttribute processContents="skip"/>
</xs:complexType>
<xs:complexType name="medialevel_qoeMetricsType">
    <xs:attribute name="sessionId" type="xs:string"/>
    <xs:attribute name="totalCorruptionDuration" type="unsignedLongVectorType"
        use="optional"/>
    <xs:attribute name="numberOfCorruptionEvents" type="unsignedLongVectorType"
        use="optional"/>
    <xs:attribute name="t" type="xs:boolean" use="optional"/>
    <xs:attribute name="totalNumberofSuccessivePacketLoss" type="unsignedLongVectorType"
        use="optional"/>
    <xs:attribute name="numberOfSuccessiveLossEvents" type="unsignedLongVectorType"
        use="optional"/>
    <xs:attribute name="numberOfReceivedPackets" type="unsignedLongVectorType"
        use="optional"/>
    <xs:attribute name="framerateDeviation" type="doubleVectorType" use="optional"/>
    <xs:attribute name="totalJitterDuration" type="doubleVectorType" use="optional"/>
    <xs:attribute name="numberOfJitterEvents" type="unsignedLongVectorType" use="optional"/>
    <xs:attribute name="framerate" type="doubleVectorType" use="optional"/>
    <xs:attribute name="codecInfo" type="stringVectorType" use="optional"/>
    <xs:attribute name="codecProfileLevel" type="stringVectorType" use="optional"/>
    <xs:attribute name="codecImageSize" type="stringVectorType" use="optional"/>
    <xs:attribute name="averageCodecBitrate" type="doubleVectorType" use="optional"/>
    <xs:anyAttribute processContents="skip"/>
</xs:complexType>
<xs:simpleType name="doubleVectorType">
    <xs:list itemType="xs:double"/>
</xs:simpleType>
<xs:simpleType name="unsignedLongVectorType">
    <xs:list itemType="xs:unsignedLong"/>
</xs:simpleType>
<xs:simpleType name="stringVectorType">
    <xs:list itemType="xs:string"/>
</xs:simpleType>
</xs:schema>
```

9.5.3.1 Use of Specific Values

Void

9.5.3.2 Example XML for the Reception Report Request

```
<?xml version="1.0" encoding="UTF-8"?>
<receptionReport xmlns="urn:3gpp:metadata:2008:MBMS:receptionreport"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:3gpp:metadata:2008:MBMS:receptionreport receptionreport.xsd">
    <receptionAcknowledgement>
        <fileURI>http://www.example.com/mbms-files/file1.3gp</fileURI>
        <fileURI>http://www.example.com/mbms-files/file2.3gp</fileURI>
        <fileURI>http://www.example.com/mbms-files/file4.3gp</fileURI>
    </receptionAcknowledgement>
</receptionReport>
```

A second example shows a statistical report for a streaming session. Note that the cell used during the second measurement period is the same cell as was used during the first period (indicated by the "=" sign).

```
<?xml version="1.0" encoding="UTF-8"?>
<receptionReport xmlns="urn:3gpp:metadata:2008:MBMS:receptionreport"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:3gpp:metadata:2008:MBMS:receptionreport receptionreport.xsd">
    <statisticalReport
        clientId="clientID"
        sessionType="streaming"
        serviceURI="bmsc.example.com"
        serviceId="serviceID">
        <qoeMetrics
            numberOfRebufferingEvents="0 1 0"
            initialBufferingDuration="3.213"
            totalRebufferingDuration="0 1.23 0"
            contentAccessTime="2.621"
            sessionStartTime="1219322514"
            sessionStopTime="1219322541"
            networkResourceCellId="240012AF134EA = 240012AF1325E">
            <medialevel_qoeMetrics
                sessionId="10.50.65.30:5050"
                framerateDeviation="0.345 0.250 0.123"
                t="false"
                numberOfSuccessiveLossEvents="5 0 3"
                numberOfCorruptionEvents="6 5 2"
                numberOfJitterEvents="0 1 0"
                totalCorruptionDuration="152 234 147"
                totalNumberofSuccessivePacketLoss="25 0 6"
                numberOfReceivedPackets="456 500 478"
                codecInfo="H263-2000/90000 = ="
                codecProfileLevel="profile=0;level=45 = ="
                codecImageSize="176x144 = ="
                averageCodecBitRate="124.5 128.0 115.1"
                totalJitterDuration="0 0.346 0"/>
        </qoeMetrics>
    </statisticalReport>
</receptionReport>
```

10 Media codecs and formats

10.1 General

The set of media decoders that are supported by the MBMS Client to support a particular media type are defined below. Speech, Audio, Video, Timed Text and Scene description media decoders are relevant for both MBMS Download and Streaming delivery. Other media decoders are only relevant for MBMS Download delivery.

10.2 Speech

If speech is supported, the AMR decoder, as specified in 3GPP TS 26.071 [48], 3GPP TS 26.090 [49], 3GPP TS 26.073 [50] and 3GPP TS 26.107 [51], shall be supported for narrow-band speech. The AMR wideband speech decoder, 3GPP TS 26.171 [52], 3GPP TS 26.190 [53], 3GPP TS 26.173 [54] and 3GPP TS 26.204 [55], shall be supported when wideband speech working at 16 kHz sampling frequency is supported.

10.3 Audio

If audio is supported, then the following two audio decoders should be supported:

- Enhanced aacPlus, as specified in 3GPP TS 26.401 [28], 3GPP TS 26.410 [29] and 3GPP TS 26.411 [30].
- Extended AMR-WB, as specified in 3GPP TS 26.290 [24], 3GPP TS 26.304 [25] and 3GPP TS 26.273 [26].

Specifically, based on the audio codec selection test results, Extended AMR-WB is strong for the scenarios marked with blue, Enhanced aacPlus is strong for the scenarios marked with orange, and both are strong for the scenarios marked with green colour in table 1.

Table 1

Content type Bit rate	Music	Speech over Music	Speech between Music	Speech
14 kbps mono				
18 kbps stereo				
24 kbps stereo				
24 kbps mono				
32 kbps stereo				
48 kbps stereo				

More recent information on the performance of the codecs based on more recent versions of the codecs can be found in TR 26.936 [86].

10.4 Synthetic audio

If synthetic audio is supported, the Scalable Polyphony MIDI (SP-MIDI) content format defined in Scalable Polyphony MIDI Specification [56] and the device requirements defined in Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP [57] should be supported.

SP-MIDI content is delivered in the structure specified in Standard MIDI Files 1.0 [58], either in format 0 or format 1.

In addition the Mobile DLS instrument format defined in [59] and the Mobile XMF content format defined in [60] should be supported.

A PSS client supporting Mobile DLS shall meet the minimum device requirements defined in [59] in section 1.3 and the requirements for the common part of the synthesizer voice as defined in ISO/IEC 10646-1 [70] in section 1.2.1.2. If Mobile DLS is supported, wavetables encoded with the G.711 A-law codec (wFormatTag value 0x0006, as defined in [59]) shall also be supported. The optional group of processing blocks as defined in [59] may be supported. Mobile DLS resources are delivered either in the file format defined in ISO/IEC 10646-1 [70], or within Mobile XMF as defined in [60]. For Mobile DLS files delivered outside of Mobile XMF, the loading application should unload Mobile DLS instruments so that the sound bank required by the SP-MIDI profile [57] is not persistently altered by temporary loadings of Mobile DLS files.

Content that pairs Mobile DLS and SP-MIDI resources is delivered in the structure specified in Mobile XMF [60]. As defined in [60], a Mobile XMF file shall contain one SP-MIDI SMF file and no more than one Mobile DLS file. PSS clients supporting Mobile XMF must not support any other resource types in the Mobile XMF file. Media handling behaviours for the SP-MIDI SMF and Mobile DLS resources contained within Mobile XMF are defined in [60].

10.5 Video

If video is supported, H.264 (AVC) Constrained Baseline Profile Level 1.3 decoder (ITU-T Recommendation H.264| ISO/IEC 14496-10 [43]) shall be supported.

In addition, if HDTV video content at a resolution of 1280x720 (720p) with progressive scan at 30 frames per second is supported by the MBMS client, H.264 (AVC) Progressive High Profile Level 3.1 decoder [43]shall be supported, wherein the maximum VCL Bit Rate shall be constrained to 14Mbps by cpbBrVclFactor & cpbBrNalFactor being fixed to 1000 and 1200 respectively.

In addition, if a MBMS client supports stereoscopic 3D video, it should support frame-packed stereoscopic 3D video with the following characteristics:

- The bitstream conforms to H.264 (AVC) Constrained Baseline Profile Level 1.3 decoder (ITU-T Recommendation H.264| ISO/IEC 14496-10 [43]), or conforms to H.264 (AVC) Progressive High Profile Level 3.1 decoder [43]. The Maximum VCL Bit Rate shall be constrained to 14Mbps by cpbBrVclFactor & cpbBrNalFactor being fixed to 1000 and 1200 respectively.

- Frame packing type is indicated by the frame packing arrangement SEI messages of H.264 (AVC) [43] as follows:
    - The syntax element frame_packing_arrangement_type has one of the defined values: 3 for Side-by-Side, 4 for Top-and-Bottom.
    - The syntax element quincunx_sampling_flag is equal to 0;
    - The syntax element content_interpretation_type is equal to 1;
    - The syntax elements spatial_flipping_flag is equal to 0;
    - The syntax element field_views_flag is equal to 0;
    - The syntax element current_frame_is_frame0_flag is equal to 0;
- When an access unit contains a frame packing arrangement SEI message A and the access unit is neither an IDR access unit nor an access unit containing a recovery point SEI message, the following two constraints apply:
    - There shall be another access unit that precedes the access unit in both decoding order and output order and that contains a frame packing arrangement SEI message B.
    - The two frame packing arrangement SEI messages A and B shall have the same value for the syntax element frame_packing_arrangement_type.

In addition, if a MBMS client supports frame-packed stereoscopic 3D video, it shall support parsing of frame packing arrangement SEI messages as specified in H.264 (AVC) [43].

Note that MBMS does not offer dynamic negotiation of media codecs.

NOTE: a High Profile decoder is able to decode a Main Profile stream that is progressively encoded.

When H.264 (AVC) is in use in the MBMS streaming delivery method, it is recommended to transmit H.264 (AVC) parameter sets within the SDP description of a stream (using sprop-parameter-sets MIME/SDP parameter [35]), and it is not recommended to transmit parameter sets within the RTP stream. Moreover, it is not recommended to reuse any parameter set identifier value that appeared previously in the SDP description or in the RTP stream. However, if a sequence parameter set is taken into use or updated within the RTP stream, it shall be contained at least in each IDR access unit and each access unit including a recovery point SEI message in which the sequence parameter set is used in the decoding process. If a picture parameter set is taken into use or updated within the RTP stream, it shall be contained at the latest in the first such access unit in each entry sequence that uses the picture parameter set in the decoding process, in which an entry sequence is defined as the access units between an IDR access unit or an access unit containing a recovery point SEI message, inclusive, and the next access unit, exclusive, in decoding order, which is either an IDR access unit or contains a recovery point SEI message.

There are no requirements on output timing conformance (annex C of ITU-T Recommendation H.264 [43]) for MBMS clients.

The H.264 (AVC) decoder in an MBMS client shall start decoding immediately when it receives data (even if the stream does not start with an IDR access unit) or alternatively no later than it receives the next IDR access unit or the next recovery point SEI message, whichever is earlier in decoding order. Note that when the interleaved packetization mode of H.264 (AVC) is in use, de-interleaving is normally done before starting the decoding process. The decoding process for a stream not starting with an IDR access unit shall be the same as for a valid H.264 (AVC) bitstream. However, the client shall be aware that such a stream may contain references to pictures not available in the decoded picture buffer.

10.6 Still images

If still images are supported, ISO/IEC JPEG [61] together with JFIF [62] decoders shall be supported. The support for ISO/IEC JPEG only applies to the following two modes:

- baseline DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF0' in 3GPP TS 26.273 [26];

- progressive DCT, non-differential, Huffman coding, as defined in table B.1, symbol 'SOF2' 3GPP TS 26.273 [26].

10.7 Bitmap graphics

If bitmap graphics is supported, the following bitmap graphics decoders should be supported:

- GIF87a, [63];
- GIF89a, [64];
- PNG, [65].

10.8 Vector graphics

If vector graphics is supported, SVG Tiny 1.2 [66], [67] and ECMAScript [68] shall be supported.

NOTE 1: The compression format for SVG content is GZIP [42], in accordance with the SVG specification [66].

NOTE 2 Content creators of SVG Tiny 1.2 are strongly recommended to follow the content creation guidelines provided in annex L of 3GPP TS 26.234 [47].

NOTE 3: A DIMS client is capable of processing SVG Tiny 1.2 data..

10.9 Text

The text decoder is intended to enable formatted text in a SMIL presentation.

If text is supported, a MBMS client shall support

- text formatted according to XHTML Mobile Profile [69];
- rendering a SMIL presentation where text is referenced with the SMIL 2.0 "text" element together with the SMIL 2.0 "src" attribute.

If text is supported, the following character coding formats shall be supported:

- UTF-8, [71];
- UCS-2, [70].

NOTE: Since both SMIL and XHTML are XML based languages it would be possible to define a SMIL plus XHTML profile. In contrast to the presently defined SMIL Language Profile that only contain SMIL modules, such a profile would also contain XHTML modules. No combined SMIL and XHTML profile is specified for MBMS. Rendering of such documents is out of the scope of the present document.

10.10 Timed text

If timed text is supported, MBMS clients shall support 3GPP TS 26.245 [72]. Timed text may be transported over RTP or downloaded contained in 3GP files using Basic profile.

NOTE: When a MBMS client supports timed text it needs to be able to receive and parse 3GP files containing the text streams. This does not imply a requirement on MBMS clients to be able to render other continuous media types contained in 3GP files, e.g. AMR, if such media types are included in a presentation together with timed text. Audio and video are instead streamed to the client using RTP.

10.11 3GPP file format

An MBMS client shall support the Basic profile and the Extended presentation profile of the 3GPP file format 3GPP TS 26.244 [32].

For delivery of 3GP-DASH formatted segments over MBMS download (see clause 5.6), the MBMS client shall support the 3GPP file format and segments for Dynamic Adaptive Streaming over HTTP as specified in 3GPP TS 26.247 [98] and in 3GPP TS 26.244 [32].

10.12 Scene Description

If scene description is supported, MBMS clients shall support 3GPP DIMS TS 26.142 [89].

10.13 Timed graphics

If timed graphics is supported, MBMS clients shall support 3GPP TS 26.430[95].

11 MBMS Metadata

11.1 The MBMS Metadata Envelope

11.1.1 Supported Metadata Syntaxes

The MBMS metadata syntax supports the following set of features:

- Support of carriage of SDP descriptions, and SDP is expected to sufficiently describe at least: MBMS Streaming sessions and, MBMS download sessions.
- Support for multiple metadata syntaxes, such that the delivery and use of more than one metadata syntax is possible.
- Consistency control of metadata versions, between senders and receivers, independent of the transport and bearer use for delivery.
- Metadata fragments are identified, versioned and time-limited (expiry described) in a metadata fragment syntax-independent manner (which is a consequence of the previous two features).

11.1.2 Consistency Control and Syntax Independence

The *metadata envelope* provides information to identify, version and expire each of its metadata fragments. This is specified to be independent of metadata fragments syntax and of transport method (thus enabling the use of more than one syntaxes and enable delivery over more than a single transport and bearer).

A metadata envelope (as identified by the *metadataEnvelope* element in the schema in sub-clause 11.1.3) consists of one or more metadata envelope instances (as identified by the *item* element in the schema in subclause 11.1.3). A metadata envelope instance is associated to exactly one metadata fragment. A metadata envelope instance may update the time validity of its metadata fragment without changing version of that metadata fragment if the metadata fragment has not changed. A newer version (higher version number) of a metadata envelope instance shall automatically expire the earlier version. If the content type (metadata fragment syntax) is recognized and valid, the UE shall use the new metadata fragment description. However, if the content type is not recognized or valid, the UE may maintain the expired version data until the newer version is correctly received.

Service announcement senders shall increment the version by one for each subsequent transported version of a metadata fragment. However, a UE shall also accept versions with an increment greater than one (so that they do not fail in the case that an intermediate version was not successfully transported).

11.1.3 Metadata Envelope Definition

The attributes for a metadata envelope instance and their description is as follows. These attributes shall be supported:

- *metadataURI*: A URI providing a unique identifier for the metadata fragment. The *metadataURI* attribute shall be present.
- *version*: The version number of the associated instance of the metadata fragment. The version number should be initialized to one. The version number shall be increased by one whenever the metadata fragment is updated. The *version* attribute shall be present.
- *validFrom*: The date and time from which the metadata fragment file is valid. The *validFrom* attribute may or not be present. If not present, the UE should assume the metadata fragment version is valid immediately.
- *validUntil*: The date and time when the metadata fragment file expires. The *validUntil* attribute may or not be present. If not present the UE should assume the associated metadata fragment is valid for all time, or until it receives a newer metadata envelope for the same metadata fragment describing a validUntil value.
- *contentType*: The MIME type of the metadata fragment which shall be used as defined for "Content-Type" in RFC 2616 [18]. The *contentType* attribute shall be present for embedding metadata envelopes. The *contentType* attribute may be present for referencing metadata envelopes.

The metadata envelope is instantiated using an XML structure. This XML contains a URI referencing the associated metadata fragment. The formal schema for the metadata envelope is defined as an XML Schema as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:3gpp:metadata:2005:MBMS:envelope"
elementFormDefault="qualified" attributeFormDefault="unqualified"
targetNamespace="urn:3gpp:metadata:2005:MBMS:envelope">
    <xs:element name="metadataEnvelope" type="metadataEnvelopeType"/>
    <xs:complexType name="metadataEnvelopeType">
        <xs:sequence>
            <xs:element name="item" type="metadataEnvelopeItemType" maxOccurs="unbounded"
minOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="metadataEnvelopeItemType">
        <xs:sequence>
            <xs:element name="metadataFragment" type="xs:string" minOccurs="0" maxOccurs="1"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="metadataURI" type="xs:anyURI" use="required"/>
        <xs:attribute name="version" type="xs:positiveInteger" use="required"/>
        <xs:attribute name="validFrom" type="xs:dateTime" use="optional"/>
        <xs:attribute name="validUntil" type="xs:dateTime" use="optional"/>
        <xs:attribute name="contentType" type="xs:string" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
</xs:schema>
```

The *metadataFragment* element shall be encapsulated in the metadata envelope for embedded metadata fragments, and shall not be encapsulated where the metadata fragment is not embedded. In the embedded case, the *metadataFragment* element shall contain exactly one embedded metadata fragment as specified by the metadata envelope syntax and only one instance of the envelope element shall be used for encapsulating envelopes.

An embedded metadata fragment (in the *metadataFragment* element) shall be escaped. Generally, an embedded metadata fragment should be escaped by placing inside a CDATA section [31]. Everything starting after "<![CDATA[" string and ending at the "]]>" string would be ignored by the XML envelope parser (quotes not included). Thus, the embedded parts would appear as "<![CDATA[" + metadata_fragment + "]]>". In this case, the complete metadata envelope with embedded metadata fragment shall not violate the rules of CDATA section usage [31].

In case the embedded metadata fragment is an XML document and include a CDATA section , the embedded metadata fragment may be escaped by replacing illegal characters with their ampersand-escaped equivalents [31] (instead of encapsulating the whole fragment in a CDATA section). For instance "<" is an illegal character that would be replaced by "<". This method is useful to avoid nesting CDATA sections (which is not allowed).

A metadata fragment which does not adhere to either of these two methods shall not be embedded in a metadata envelope, thus it may only be referenced from an referencing metadata envelope.

Embedded fragments are not expected to be parsed by the metadata envelope XML parser, but decapsulated and passed to the relevant metadata management operation that is implementation specific (e.g. for immediate parsing, storage, etc.).

11.1.4 Delivery of the Metadata Envelope

An instance of metadata envelope shall be associated with an instance of a metadata fragment by one of two methods:

- Embedded: The metadata fragment is embedded within the metadata envelope.
- Referenced: The metadata fragment is referenced from the metadata envelope.

The MBMS UE must know the MIME Type of each metadata fragment.

In the embedded case, the envelope and fragment are, by definition, transported together and in-band of one another. In the referenced case, the envelope and fragment shall be transported together in-band of the same transport session.

MBMS Service Announcement transports shall support delivery of the metadata envelope as a discrete object (XML document) for the referenced case. In the referenced case, the MIME type of the metadata fragment should be provided by the transport protocol (e.g. as a Content-Type text string). In both cases, the MIME type of the metadata envelope should be provided by the transport protocol.

The metadata envelope instance includes a reference (*metadataURI*) to the associated metadata fragment using the same URI as the fragment file is identified by in the Service Announcement. Thus, metadata envelope can be mapped to its associated metadata fragment.

11.2 MBMS User Service Description Metadata Fragment

11.2.1 Definition of the MBMS User Service Bundle Description

11.2.1.1 Initial Definition

The root element of the MBMS User Service Bundle description is the *bundleDescription* element. The element is of the *bundleDescriptionType*. The *bundleDescription* contains one or several *userServiceDescription* elements and optionally a reference to the FEC repair stream description.

Each *userServiceDescription* element shall have a unique identifier. The unique identifier shall be offered as *serviceId* attribute within the *userServiceDescription* element and shall be of URN format.

The *userServiceDescription* element may contain one or more *name* elements. The intention of a *name* element is to offer a title of the user service. For each name elements, the language shall be specified according to XML datatypes (XML Schema Part 2 [22]).

The *userServiceDescription* element may contain one or more *serviceLanguage* elements. Each *serviceLanguage* element represents the available languages of the user services. The language shall be specified according to XML datatypes (XML Schema Part 2 [22]).

The *userServiceDescription* element may contain an *accessPointName* attribute. The *accessPointName* attribute is optional and gives an Access Point Name (APN) as defined in [77]. When this attribute is present, the MBMS UE shall use the given APN for MBMS UE to network interactions like File Repair and/or security registration. If this attribute is not present, the MBMS UE shall use a default PDP context/default EPS bearer for network interactions.

Each *userServiceDescription* element shall contain at least one *deliveryMethod* element. The *deliveryMethod* element contains the description of one delivery method. The element shall contain one reference to a session description and may contain references to one associated delivery procedure and/or one service protection descriptions. The session description is further specified in sub-clause 5.2.2.2.

A *requiredCapabilities* element gives a list of features, which are required for the consumption of the related MBMS user service. The list of features that are currently defined is included in section 11.9. The value of the *feature* element indicates the required feature. Note that the BM-SC can also determine the terminal capabilities from the terminal

identification during the security registration. If the registering terminal does not have the required capabilities, the BM-SC can reject the security registration.

The *deliveryMethod* element may contain a reference to an associated delivery procedure description. The description and configuration of associated delivery procedures is specified in sub-clause 5.2.2.3.

The *deliveryMethod* element may contain a reference to a service protection description. The service protection description is specified in sub-clause 5.2.2.4.

A *userServiceDescription* element contains zero or more *accessGroup* elements. An *accessGroup* element defines a list of access networks and is uniquely identified by its id attribute. An *accessGroup* element describes whether separate access systems for the same MBMS user service are used (see sub-clause 5.1.5.2 of [4]) by including one or more *accessBearer* elements, each describing one of those access systems and no two describing the same. Possible *accessBearer* values are “3GPP.R6.UTRAN”, “3GPP.R6.GERAN” , “3GPP.R7.MBSFN-FDD”, “3GPP.R7.MBSFN-TDD” and “3GPP.R8.MBSFN-IMB” which indicate transport by 3GPP MBMS bearers according to the specification in [4][5]. The *accessBearer* value for evolved UTRAN is “3GPP.R9.E-UTRAN”.

For forward compatibility, other *accessBearer* values are allowed but their definition and use are out of scope of this specification and a 3GPP UE may silently ignore other values.

Each *deliveryMethod* element contains at most one *accessGroupId* attribute. One specific *accessGroupId* value maps to one specific *accessGroup* element id value. For each unique *accessGroupId* attribute value presented in a *deliveryMethod* element of a *userServiceDescription* instance, exactly one associated *accessGroup* element shall be present and the id attribute of the *accessGroup* element and the *accessGroupId* attribute shall have the same value. For each *deliveryMethod* element without an *accessGroupId* attribute, the UE should assume that the delivery method is offered through all available MBMS access systems.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3GPP:metadata:2005:MBMS:userServiceDescription" elementFormDefault="qualified">
    <xs:element name="bundleDescription" type="bundleDescriptionType"/>
    <xs:complexType name="bundleDescriptionType">
        <xs:sequence>
            <xs:element name="userServiceDescription" type="userServiceDescriptionType"
maxOccurs="unbounded"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="fecDescriptionURI" type="xs:anyURI" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="userServiceDescriptionType">
        <xs:sequence>
            <xs:element name="name" type="nameType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="serviceLanguage" type="xs:language" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="requiredCapabilities" type="requirementsType" minOccurs="0"
maxOccurs="1"/>
            <xs:element name="deliveryMethod" type="deliveryMethodType" maxOccurs="unbounded"/>
            <xs:element name="accessGroup" type="accessGroupType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="serviceId" type="xs:anyURI" use="required"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="accessGroupType">
        <xs:sequence>
            <xs:element name="accessBearer" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="accessGroupIdType" use="required"/>
    </xs:complexType>
    <xs:complexType name="deliveryMethodType">
        <xs:sequence>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="accessGroupId" type="accessGroupIdType" use="optional"/>
        <xs:attribute name="associatedProcedureDescriptionURI" type="xs:anyURI" use="optional"/>
        <xs:attribute name="protectionDescriptionURI" type="xs:anyURI" use="optional"/>
        <xs:attribute name="sessionDescriptionURI" type="xs:anyURI" use="required"/>
        <xs:attribute name="accessPointName" type="xs:anyURI" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
```

```
    <xs:complexType name="nameType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="lang" type="xs:language" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="accessGroupIdType">
        <xs:restriction base="xs:nonNegativeInteger">
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="requirementsType">
        <xs:sequence>
            <xs:element name="feature" type="xs:unsignedInt" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Note that Annex I contains an informative User Service Description and extensions schema.

11.2.1.2 Extensions to the User Service Bundle Description

The MBMS User Service Bundle Description schema defined in this clause extends the MBMS Release 6 schema of clause 11.2.1.1. An MBMS User Service Description schema of the current release shall comply with MBMS User Service Description schema definition of Release 6 and the subsequent releases up to the current release.

An *initiationRandomization* element and *terminationRandomization* element carries the parameters to be used by the MBMS UE to randomize their initiation and/or termination operations over time. If the *initiationRandomization* element is present, all MBMS UEs shall randomize the initiation time as defined by the attributes of the elements. If the *terminationRandomization* element is present, all MBMS UEs shall randomize the termination time as defined by the attributes of the elements.

The *initiationRandomization* and/or *terminationRandomization* element may be part of:

- a bundleDescription, where it applies to all services in the service bundle
- a userServiceDescription, where it applies to all MBMS bearer services of a single service. If present, this overrides the element in bundleDescription

If the *initiationRandomization* element is not present, the MBMS UE does not randomize the User Service Initiation procedure over time. The MBMS UE should then perform the operation immediately when it is triggered.

If the *terminationRandomization* element is not present, the MBMS UE does not randomize the User Service Termination procedure over time. The MBMS UE should then perform the operation immediately when it is triggered.

An *initiationRandomization* element may contain the *initiationStartTime* attribute, which defines the start time for the initiation procedure randomization period. The value of the data field represents the 32 most significant bits of a 64 bit Network Time Protocol (NTP) [78] time value. If the *initiationStartTime* attribute is not present, the MBMS UE shall use the reception time of the User Service Discovery / Announcement information as *initiationStartTime*.

The *initiationRandomization* element shall contain the *protectionPeriod* attribute. The *protectionPeriod* attribute expresses the length of the protection period in seconds. The initiation procedure shall be randomly deferred during protection period.

The *initiationRandomization* element shall contain the *randomTimePeriod* attribute. The *randomTimePeriod* attribute expresses the length of a time interval (in seconds) over which requests are deferred. The MBMS UE shall calculate a random time for the execution of the initiation procedure. The method provides for statistically uniform distribution over a relevant period of time.

The *terminationRandomization* element shall contain the *protectionPeriod* attribute. The *protectionPeriod* attribute expresses the length of the protection period in seconds. The termination procedure execution shall be randomly deferred during protection period.

The *terminationRandomization* element shall contain the *randomTimePeriod* attribute. The *randomTimePeriod* attribute expresses the length of a time interval (in seconds) over which the operations are deferred. The MBMS UE shall calculate a random time for the execution of the termination procedure. The method provides for statistically uniform distribution over a relevant period of time.

If the MBMS UE is switched off during the termination randomization, the MBMS UE shall cancel the termination randomization.

The *alternativeAccessDelivery* element shall extend the list of elements of the MBMS *deliveryMethod* element. Whenever present, it shall contain at least one unicastAccessURI element. The *unicastAccessURI* element provides unicast server information for OMA push for MBMS download service when the UE is outside of home network and file download delivery method is used. The *unicastAccessURI* element refers to a URI to be used for unicast access to the streaming service. If the alternativeAccessDelivery element is available then the UE shall select one of the unicastAccessURI elements included.The timeShiftingBuffer attribute may be used to indicate the minimal size of the time shifting buffer that will be provided for the current service by the PSS servers that are referenced in the list. The actual size of the timeshifting buffer of the selected server is returned in the SETUP response from the PSS Server.

A USD may include a *r7:unicastAccessURI* element for support of Release 7 UEs. UEs of release 8 onwards shall use *alternativeAccessDelivery* element for both OMA Push file download and unicast streaming.

The *serviceClass* attribute, if present, shall extend the list of attributes of the MBMS Release 6 *userServiceDescription* element. The *serviceClass* attribute is optional and contains the service class identifier for the delivered service according to the syntax defined in clause E.1.2 of [90]. Note that Annex E of [90] also foresees the registration of service class identifiers with the Open Mobile Naming Authority. The service class identifier is similar to MIME types and provides an unique identity to services. A MBMS UE may determine the receiving application instance out of the service class identifier.

A user service description may belong to a group of user service descriptions, which represent alternative configurations of the same user service. An example is an MBMS user service that is delivered over non-MBSFN bearer with a low bitrate and over MBSFN bearer with a high bitrate. In such a case, the UE is only expected to consume one variant of the service. The UE recognizes that a set of user service descriptions apply to one user service based on the *serviceGroup* element.

The MBMS User Service Description may include a *Registration* element. If present, then the UE shall send the registration and deregistration to the given URI. In such a case, the User Service Description fragment may not be complete in the service announcement. Instead, it may contain references to metadata fragments (e.g. the session description) that are not embedded in the service announcement. When registration is requested, the userServiceDescription element shall contain a Registration element that describes the requested registration procedure. The registrationURI indicates the URI to the server with whom the registration procedure shall be performed. In case more than one registrationURI is indicated, the UE shall select one randomly. The registrationThreshold is a number that indicates the percentage of UEs that are requested to register. The UE shall randomly select a number between 0 and 100 and compare it against the threshold. In case the selected number is lower than the indicated threshold, the UE should perform registration. The threshold value "100" indicates that the UE shall perform registration, which is e.g. necessary when the USD is not complete.

The MBMS User Service Description may include a *mediaPresentationDescription* element, when the MBMS User Service Description describes a download delivery service. If present, then the UE expects that the received files correspond to a media presentation described by the MPD in [98]. The *mpdURI* element provides a reference to the Media Presentation Description metadata fragment, whose contents are identical to the MPD as defined in [98]. The Media Presentation Description fragment may refer, via the InitialisationSegmentURL, to one or more Initialisation Segment Description (ISD) metadata fragments, whose contents are identical to the Initialisation Segment as defined in [98].

The MBMS User Service Description may include a *schedule* element. If present, the *schedule* element includes a URI reference to the MBMS User Service schedule information, the latter corresponding to a Schedule Description metadata fragment as defined in sub-clause 11.2A.

The Schedule Description fragment may include a URI reference to the Filter Description metadata fragment to signify the intended use of content filtering. Each filter data instance contains associated criteria or rules for use by UEs to selectively receive contents at the session level or the file level. At the session level, the filter rules enable a UE to decide whether it should receive the entire contents of the associated delivery session(s) of the User Service. At the file level, the filter rules enable a UE to decide whether it should receive individual files of the User Service, during the corresponding file delivery schedule(s). In this release of the specification, the Filter Description fragment contains strictly location filtering data as defined by the *locationFilter* element. Additional description of the association between Schedule Description and Filter Description fragments and intended usage are given in sub-clauses 11.2A and 11.2B.

The MBMS User Service Description may include an *availabilityInfo* element. If present, it shall extend the list of child elements of the MBMS Release 11 *userServiceDescription* element by indicating the presence of additional data pertaining to the availability of the service.

The *availabilityInfo* element shall include one or more *infoBinding* elements. The *infoBinding* element shall contain the child elements *serviceArea* and *radiofrequency*. A UE shall be capable of processing an *infoBinding* element that does not contain the child element *serviceArea.* Note that for backwards compatibility reasons, *serviceArea* needs to be indicated as optional in the USD schema (i.e. 'minOccurs="0"'). The *serviceArea* element declares the one or more service areas over which this MBMS User Service is provided, This element is designated by the *MBMS Service Area Identity* as defined in 3GPP TS 36.443 [104] and 3GPP TS 23.003 [77]. According to 3GPP TS 36.443 [104], *MBMS Service Area Identity* is frequency agnostic and can be mapped onto one or more cells. The specific usage of the *MBMS Service Area Identity*, or its correlation to other network identification information, is not defined in this specification. The *radioFrequency* element indicates the one or more RF frequencies in the E-UTRAN downlink which transmit this MBMS User Service over the service area(s) identified by the *serviceArea* element. The frequency parameter is coded as EARFCN in 3GPP TS 36.101 [105]. The MBMS client shall forward the service area and radio frequency information received in the USD to the lower layers, and the UE is expected to make use of such information in accordance with TS 36.300 [96] clause 15.4 as well as TS 36.304 [108] and TS 36.331 [97].

The following schema defines the Release 7 extensions to the User Service Bundle Description schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2007:MBMS:userServiceDescription"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3GPP:metadata:2007:MBMS:userServiceDescription"
elementFormDefault="qualified">
    <xs:element name="initiationRandomization">
        <xs:complexType>
            <xs:attribute name="initiationStartTime" type="xs:unsignedInt" use="optional"/>
            <xs:attribute name="protectionPeriod" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="randomTimePeriod" type="xs:unsignedInt" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="terminationRandomization">
        <xs:complexType>
            <xs:attribute name="protectionPeriod" type="xs:unsignedInt" use="required"/>
            <xs:attribute name="randomTimePeriod" type="xs:unsignedInt" use="required"/>
        </xs:complexType>
    </xs:element>
    <xs:element name="serviceGroup" type="serviceGroupType"/>
    <xs:complexType name="serviceGroupType">
        <xs:attribute name="groupID" type="xs:anyURI" use="required"/>
    </xs:complexType>
    <xs:element name="unicastAccessURI" type="xs:anyURI"/>
    <xs:attribute name="serviceClass" type="xs:string"/>
</xs:schema>
```

The following schema defines the Release 8 extensions to the User Service Bundle Description schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2008:MBMS:userServiceDescription"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3GPP:metadata:2008:MBMS:userServiceDescription"
    elementFormDefault="qualified">

    <xs:element name="alternativeAccessDelivery">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="unicastAccessURI" type="xs:anyURI" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="timeShiftingBuffer" type="xs:unsignedInt" use="optional"
default="0"/>
        </xs:complexType>
    </xs:element>

    <xs:element name="Registration">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="registrationURI" type="xs:anyURI" minOccurs="1"
   maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="registrationThreshold" type="xs:unsignedInt" use="optional"
   default="100"/>
```

```
        </xs:complexType>
    </xs:element>
</xs:schema>
```

The following schema defines the Release 9 extensions to the User Service Bundle Description schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2009:MBMS:userServiceDescription"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3GPP:metadata:2009:MBMS:userServiceDescription"
    elementFormDefault="qualified">

     <xs:element name="mediaPresentationDescription">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="mpdURI" type="xs:anyURI"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="schedule">
        <xs:complexType>
        <xs:sequence>
        <xs:element name="scheduleDescriptionURI" type="xs:anyURI"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="availabilityInfo">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="infoBinding" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="serviceArea" type="xs:unsignedShort" minOccurs="0"
maxOccurs="unbounded"/>
                            <xs:element name="radioFrequency" type="xs:unsignedInt"
maxOccurs="unbounded"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
</xs:element>
</xs:schema>
```

Note that Annex J contains a main User Service Description schema referencing the extensions schema.

11.2.2 Example MBMS User Service Description Instances

All examples in this sub-clause are generated out of the network supporting the current release USD schema.

The following User Service Bundle Description instance is an example of a simple fragment. This fragment includes only the mandatory elements.

```
<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription
                        USD-schema-main.xsd">
    <userServiceDescription serviceId="urn:3gpp:0010120123hotdog">
        <deliveryMethod sessionDescriptionURI="http://www.example.com/3gpp/mbms/session1.sdp">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <sv:delimiter>0</sv:delimiter>
    </userServiceDescription>
    <sv:schemaVersion>1</sv:schemaVersion>
</bundleDescription>
```

The following User Service Description instance is an example of a fuller fragment.

```
<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription
                        USD-schema-main.xsd"
    fecDescriptionURI="http://www.example.com/3gpp/mbms/session1-fec.sdp">
    <userServiceDescription serviceId="urn:3gpp:1234567890coolcat">
        <name lang="EN">Welcome</name>
        <name lang="DE">Willkommen</name>
        <name lang="FR">Bienvenue</name>
        <name lang="FI">Tervetuloa</name>
        <serviceLanguage>EN</serviceLanguage>
        <serviceLanguage>DE</serviceLanguage>
        <requiredCapabilities>
            <feature>0"</feature>
        </requiredCapabilities>
        <deliveryMethod accessGroupId="1"
            sessionDescriptionURI="http://www.example.com/3gpp/mbms/session1.sdp">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <deliveryMethod sessionDescriptionURI="http://www.example.com/3gpp/mbms/session2.sdp"
            associatedProcedureDescriptionURI="http://www.example.com/3gpp/mbms/procedureX.xml">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <deliveryMethod sessionDescriptionURI="http://www.example.com/3gpp/mbms/session3.sdp"
            associatedProcedureDescriptionURI="http://www.example.com/3gpp/mbms/procedureY.xml">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <deliveryMethod accessGroupId="2"
            sessionDescriptionURI="http://www.example.com/3gpp/mbms/session4.sdp">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <accessGroup id="1">
            <accessBearer>3GPP.R6.GERAN</accessBearer>
            <accessBearer>3GPP.R6.UTRAN</accessBearer>
        </accessGroup>
        <accessGroup id="2">
            <accessBearer>3GPP.R6.UTRAN</accessBearer>
        </accessGroup>
        <sv:delimiter>0</sv:delimiter>
    </userServiceDescription>
    <sv:schemaVersion>1</sv:schemaVersion>
</bundleDescription>
```

The following User Service Description instance is an example of a Release 7 fragment.

```
<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xmlns:r7="urn:3GPP:metadata:2007:MBMS:userServiceDescription"
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription
                        USD-schema-main.xsd"
    fecDescriptionURI="http://www.example.com/3gpp/mbms/session1-fec.sdp">
    <userServiceDescription serviceId="urn:3gpp:1234567890coolcat">
        <name lang="EN">Welcome</name>
        <name lang="DE">Willkommen</name>
        <name lang="FR">Bienvenue</name>
        <name lang="FI">Tervetuloa</name>
        <serviceLanguage>EN</serviceLanguage>
        <serviceLanguage>DE</serviceLanguage>
        <deliveryMethod accessGroupId="1"
            sessionDescriptionURI="http://www.example.com/3gpp/mbms/session1.sdp">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <deliveryMethod sessionDescriptionURI="http://www.example.com/3gpp/mbms/session2.sdp"
            associatedProcedureDescriptionURI="http://www.example.com/3gpp/mbms/procedureX.xml">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <deliveryMethod sessionDescriptionURI="http://www.example.com/3gpp/mbms/session3.sdp"
            associatedProcedureDescriptionURI="http://www.example.com/3gpp/mbms/procedureY.xml">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <deliveryMethod accessGroupId="2"
            sessionDescriptionURI="http://www.example.com/3gpp/mbms/session4.sdp">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <accessGroup id="1">
            <accessBearer>3GPP.R6.GERAN</accessBearer>
            <accessBearer>3GPP.R6.UTRAN</accessBearer>
        </accessGroup>
        <accessGroup id="2">
            <accessBearer>3GPP.R6.UTRAN</accessBearer>
        </accessGroup>
        <r7:serviceGroup groupID="http://www.example.com/mbms/serviceGroup1"/>
        <r7:initiationRandomization initiationStartTime="3468452458" protectionPeriod="600"
randomTimePeriod="300"/>
        <r7:terminationRandomization protectionPeriod="300" randomTimePeriod="120"/>
        <sv:delimiter>0</sv:delimiter>
    </userServiceDescription>
    <sv:schemaVersion>1</sv:schemaVersion>
</bundleDescription>
```

The following example User Service Description instance adds an RTSP URI for alternative access to the delivery method.

```
<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xmlns:r7="urn:3GPP:metadata:2007:MBMS:userServiceDescription">
    xmlns:r8="urn:3GPP:metadata:2008:MBMS:userServiceDescription">
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription
                        USD-schema-main.xsd">
    <userServiceDescription serviceId="urn:3gpp:1234567890MobileTVChannelBundleCh1"
    r7:serviceClass="urn:oma:bcast:ext_bsc_3gpp:example_service:1.0">
                <deliveryMethod
            sessionDescriptionURI="http://www.example.com/3gpp/mbms/channel1.sdp">
            <r8:alternativeAccessDelivery timeShiftingBuffer="3600">
            <r8:unicastAccessURI>rtsp://www.example.com/3gpp/mbms/channel1_pss.sdp
            </r8:unicastAccessURI>
            </r8:alternativeAccessDelivery>
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <sv:delimiter>0</sv:delimiter>
    </userServiceDescription>
    <sv:schemaVersion>1</sv:schemaVersion>
```

```
</bundleDescription>
```

The following example User Service Description instance indicates that a registration procedure is requested for 50% of the UEs before the consumption of the MBMS User Service.

```
<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xmlns:r7="urn:3GPP:metadata:2007:MBMS:userServiceDescription"
    xmlns:r8="urn:3GPP:metadata:2008:MBMS:userServiceDescription">
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription
                        USD-schema-main.xsd">   <userServiceDescription
serviceId="urn:3gpp:1234567890MobileTVChannelBundleCh1"
                    r7:serviceClass="urn:oma:bcast:ext_bsc_3gpp:example_service:1.0">
        <deliveryMethod sessionDescriptionURI="http://www.example.com/3gpp/mbms/channel1.sdp">
            <sv:delimiter>0</sv:delimiter>
        </deliveryMethod>
        <r8:Registration registrationThreshold="50">
                <registrationURI> http://www.example.com/3gpp/mbms/register.php</registrationURI>
        </r8:Registration>
        <sv:delimiter>0</sv:delimiter>
    </userServiceDescription>
    <sv:schemaVersion>1</sv:schemaVersion>
</bundleDescription>
```

11.2A Schedule Description Metadata Fragment

The XML schema for the Schedule Description Metadata Fragment is specified in sub-clause 11.2A.2. The procedures associated with the elements of the schema are specified in sub-clause 11.2A.1.

11.2A.1 Procedures for Schedule Description Metadata Fragment

A schedule description instance may be delivered to the MBMS clients:

- during a User Service Discovery / Announcement prior to the MBMS Download delivery session along with the session description (out-of-band of that session); or
- in-band within a MBMS Download delivery session; or
- via an MBMS download delivery session dedicated to the transport of file schedule information.

The most recently delivered schedule file shall take priority, such that schedule parameters received prior to, and out-of-band, of the download session they apply to are regarded as "initial defaults", and schedule parameters received in-band with the download session, overwrite the earlier received schedule parameters. The MBMS Metadata envelope, see sub-clause 11.1, applies for the reception of schedule parameters.

The schedule description instance is clearly identified using a URI, to enable UE cross-referencing of in and out-of-band schedule files.

The MIME Type for the Schedule information is "application/mbms-schedule+xml"

Availability of the schedule description metadata fragment is indicated by the presence of the *schedule* element in the MBMS User Service Bundle Description fragment. The URI to the Schedule Description fragment is provided by the element *scheduleDescriptionURI* in the *schedule* element.

The start and stop time of a single event is specified by the *start* and *stop* elements/attributes, which provides the absolute date and time of the start time and the stop time of the event. The duration may be determined by subtracting the start time from the stop time.

The *reoccurencePattern* element if included shall have a value of either "daily", "weekly" or "monthly".

The number of reoccurence of an event may be indicated by either specifying the end time, as indicated by the *reoccurenceStopTime* element, or by specifying the number of reoccurrence, with the *numberOfTimes* element. If there are no reoccurrence, then the *reoccurencePattern*, *numberOfTimes* and *reoccurenceStopTime* elements are not included.

The *fileSchedule* element specifies details about the files delivered during a session. The *sessionID* attribute is as defined in sub-clause 9.4.6. If present, it identifies the delivery session for each file. If not present, a UE shall determine the transport session as defined by the session description for the download session. The *fileMD5* attribute is the MD5 hash value of the file. If present, the purpose of this hash is to enable a UE to determine if a file has changed since a prior reception without having to download the file.

The *scheduleUpdate* element specifies a time after which UE shall seek to update its schedule information.

An *index* element is included as a child of sessionSchedule element. The index is the starting index in a sessionSchedule describing recurrence with the index increased by one from the index for each session in the session repetitions.

A *cancelled* attribute is defined as a child of the *fileURI* element, itself a child of the *fileSchedule* element. If *cancelled* is set to "true" or "1", then the transmission of the file identified by the *fileURI* element is cancelled, and the UE shall cancel any applicable file repair and/or reception reporting procedures for that file . If this file schedule-level cancellation indication in the updated schedule description is received after the associated file has already been delivered, then any related file repair, or reception reporting for that file (associated with its parent service), either in progress or yet to occur, shall be aborted. If *cancelled* is set to "false" or "0" or is absent, then nominal file transmission and associated delivery procedures, if applicable, shall occur.

A *sessionScheduleOverride* element is defined as a child of the *serviceSchedule* element. If included, the *sessionScheduleOverride* element indicates either the cancellation of the session occurrence, or schedule override, as follows:

- If the *cancelled* attribute (a child of *sessionScheduleOverride* element) is set to "true" or "1", then the transmission of the session identified by the *index* attribute (a child of *sessionScheduleOverride* element) is cancelled, and the UE shall cancel any applicable file repair and/or reception reporting for all files belonging to that session. If this session schedule-level cancellation indication in the updated schedule description is received after any of the associated files have already been delivered, then any related file repair, or reception reporting for those files (associated with their parent service(s)), either in progress or yet to occur, shall be aborted.

- If the *cancelled* attribute (a child of *sessionScheduleOverride* element) is set to "false" or "0" or is absent, then the *start* and *stop* time elements (children of *sessionScheduleOverride* element) shall override the nominal start and stop time of the transmission schedule of the session as identified by the *index* attribute (a child of *sessionScheduleOverride* element) .

The value of the *index* attribute in the *sessionScheduleOverride* element corresponds to any of the the value of the *index* element in the *reoccurenceStartStopType* in the *sessionSchedule* element.

Schedule information received in the Schedule Description metadata fragment shall take precedence over timing information that may have been received in SDP (t or/and r lines).

The child element *receptionFiltering* may be present in either the *sessionSchedule* or *fileSchedule* elements of the Schedule Description fragment. If it appears in the session schedule, *receptionFiltering* signifies the presence of the Filter Description metadata fragment for use by the UE to perform selective reception of contents, in their entirety, sent during the corresponding session(s) of the User Service. If it appears in the file schedule, *receptionFiltering* signifies the presence of the Filter Description metadata fragment for use by the UE to perform selective reception of the corresponding file(s) of the User Services, during its(their) scheduled delivery time(s). Should *receptionFiltering* be present in both the *sessionSchedule* and *fileSchedule* elements, *fileSchedule* shall take precedence. The *filterDescriptionReference* attribute of the Schedule Description fragment identifies the Filter Description fragment of concern, and each instance of the *data* child element of *receptionFiltering* identifies a unique filter data instance in the Filter Description fragment to be applied for content filtering and selective reception. Multiple *data* elements may appear under *receptionFiltering*, to accommodate the presence of different categories or types of filtering data. In this release of the specification, only location filtering data is defined. More details on the composition of filtering data are given in sub-clause 11.2B.

11.2A.2 XML-Schema for the Schedule Description Meta Data Fragment

Below is the formal XML syntax of schedule information procedure. Documents following this schema can be identified with the MIME type "application/mbms-schedule+xml" defined in Annex C.14. The file name of XML schema for schedule description is Schedule-Description-Main.xsd.

In this version of the specification the network shall set the *schemaVersion* element, defined as a child of *scheduleDescription* element, to 2.

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 a UE receives an instantiation of a Schedule Description compliant to this schema, it shall determine the schema version required to parse the instantiation as follows:

- If the UE supports one or more versions of the Schedule Description schema with the schema *version* attribute, then the UE 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 XML schema "schema-version.xsd" is specified in Annex J.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3gpp:metadata:2011:MBMS:scheduleDescription"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:r11="urn:3gpp:metadata:2012:MBMS:scheduleDescription"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    targetNamespace="urn:3gpp:metadata:2011:MBMS:scheduleDescription"
elementFormDefault="qualified"
    version="2">
<xs:import schemaLocation="Schedule-Rel-11-schema-snippet.xsd"
namespace="urn:3gpp:metadata:2012:MBMS:scheduleDescription"/>
<xs:import schemaLocation="schema-version.xsd"
namespace="urn:3gpp:metadata:2009:MBMS:schemaVersion"/>
    <xs:complexType name="scheduleDescriptionType">
        <xs:sequence>
<xs:element ref="sv:schemaVersion"/>
<xs:element name="serviceSchedule" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="sessionSchedule" type="reoccurenceStartStopType"
minOccurs="0" maxOccurs="unbounded"/>
                        <xs:element name="sessionScheduleOverride" minOccurs="0"
maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence minOccurs="0">
                                    <xs:element name="start" type="xs:dateTime"/>
                                    <xs:element name="stop" type="xs:dateTime"/>
                                </xs:sequence>
                                <xs:attribute name="index" type="xs:unsignedInt" use="required"/>
                                <xs:attribute name="cancelled" type="xs:boolean"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="fileSchedule" minOccurs="0" maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="fileURI">
                                        <xs:complexType>
                                            <xs:simpleContent>
                                                <xs:extension base="xs:anyURI">
                                                    <xs:attribute name="cancelled"
type="xs:boolean"/>
                                                </xs:extension>
                                            </xs:simpleContent>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="deliveryInfo" minOccurs="0"
maxOccurs="unbounded">
    <xs:complexType>
                                            <xs:attribute name="start" type="xs:dateTime"/>
                                            <xs:attribute name="end" type="xs:dateTime"/>
                                            <xs:anyAttribute processContents="skip"/>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element ref="r11:receptionFiltering" minOccurs="0"/>
                                    <xs:element ref="sv:delimiter"/>
                                    <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
```

```
                                </xs:sequence>
                                <xs:attribute ref="r11:sessionId" use="optional"/>
                                <xs:attribute ref="r11:fileMD5" use="optional"/>
                                <xs:anyAttribute processContents="skip"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="serviceId" type="xs:anyURI"/>
                    <xs:attribute name="serviceClass" type="xs:string" use="optional"/>
<xs:anyAttribute processContents="skip"/>
                </xs:complexType>
            </xs:element>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="scheduleUpdate" type="xs:dateTime"/>
        <xs:attribute ref="r11:filterDescriptionReference"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="reoccurenceStartStopType">
        <xs:sequence>
            <xs:element name="start" type="xs:dateTime"/>
            <xs:element name="stop" type="xs:dateTime"/>
            <xs:element name="reoccurencePattern" type="xs:string" minOccurs="0"/>
            <xs:element name="numberOfTimes" type="xs:unsignedInt" minOccurs="0"/>
            <xs:element name="reoccurenceStopTime" type="xs:dateTime" minOccurs="0"/>
            <xs:element name="index" type="xs:unsignedInt" minOccurs="0"/>
            <xs:element ref="r11:receptionFiltering" minOccurs="0"/>
            <xs:element ref="sv:delimiter"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:element name="scheduleDescription" type="scheduleDescriptionType"/>
</xs:schema>

The following schema is the release 11 extension to the Schedule Description schema. The schema file
name, as referenced in the main Schedule Description schema, is Schedule-Rel-11-schema-snippet.xsd
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3gpp:metadata:2012:MBMS:scheduleDescription"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3gpp:metadata:2012:MBMS:scheduleDescription"
    elementFormDefault="qualified">

    <xs:attribute name="sessionId" type="xs:string"/>
    <xs:attribute name="fileMD5" type="xs:base64Binary"/>

    <xs:attribute name="filterDescriptionReference" type="xs:anyURI"/>
    <xs:element name="receptionFiltering">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="data" maxOccurs="unbounded">
                    <xs:complexType>
                        <xs:attribute name="filterID" type="xs:anyURI" use="required"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

</xs:schema>
```

11.2A.3 Examples of the Schedule Description Meta Data Fragment

Example 1

In this instantiation example, the following key points can be noted:

- The version of the schema used to generate this instantiation is version 2, as indicated by the schemaVersion element. Thus a receiver of this instantiation that has multiple schedule schema versions (say 1, 2 and 3), then it should use a schedule schema version equals to 2, as indicated in the version attribute of the schema

instruction "schema". A receiver that has only version 1 of the schedule schema version should use that schema for validating the instantiation, but extensions made in version 2 of the schema will be ignored by the receiver. The receiver should avoid using schema version higher than 2, since the instantiation would fail verification against such schema version.

- The scheduleUpdate element indicates UTC time February 1st 2012 at time 00 hour 00 min 00 sec. Thus UE should seek to update the schedule instance after this time.
- There are 2 serviceSchedule elements.
- In the first serviceSchedule, it includes
  - a sessionSchedule that starts at UTC March 1st 2012 23:00:00 and ends at UTC March 1st 2012 23:30:00;
  - a first fileSchedule used for FOTA (firmware update over the air) for oem-1 model-1 with a filename of image032212.apk, which delivery starts at UTC March 1st 2012 23:00:00 and ends at March 1st 2012 23:10:00, thus 10 minutes duration;
  - a second fileSchedule used for FOTA for oem-1 model-2 with filename image098798.apk, which delivery starts at UTC March 1st 2012 23:10:00 and ends at UTC March 1st 2012 23:20:00, thus 10 minutes duration;
  - a third fileSchedule used for FOTA for oem-1 model-3 with filename image765987.apk, which delivery starts at UTC March 1st 2012 23:20:00 and ends at UTC March 1st 2012 23:30:00, thus 10 minutes duration.
- In the second serviceSchedule, it includes
  - a sessionSchedule that starts at UTC March 7th 2012 10:00:00 and ends at UTC March 7th 2012 10:30:00;
  - a first fileSchedule used for FOTA for oem-1 model-4 with a filename of image456345.apk, which delivery starts at UTC March 7th 2012 10:00:00 and ends at March 7th 2012 10:15:00, thus 15 minutes duration;
  - a second fileSchedule used for FOTA for oem-1 model-2 with filename image504123.apk, which delivery starts at UTC March 7th 2012 10:15:00 and ends at UTC March 7th 2012 10:30:00, thus 15 minutes duration.

```
<?xml version="1.0" encoding="UTF-8"?>
<scheduleDescription
    xmlns="urn:3gpp:metadata:2011:MBMS:scheduleDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:3gpp:metadata:2011:MBMS:scheduleDescription Schedule-Description-
Main.xsd"
    scheduleUpdate="2012-02-01T00:00:00Z">
        <sv:schemaVersion>2</sv:schemaVersion>
        <serviceSchedule>
           <sessionSchedule>
                <start>2012-03-01T23:00:00Z</start>
                <stop>2012-03-01T23:30:00Z</stop>
                <sv:delimiter>0</sv:delimiter>
            </sessionSchedule>
            <fileSchedule>
                <fileURI>file://fota.operator.com/swupdate/oem-1/model-1/image032212.apk</fileURI>
                <deliveryInfo start="2012-03-01T23:00:00Z" end="2012-03-01T23:10:00Z"/>
                <sv:delimiter>0</sv:delimiter>
            </fileSchedule>
            <fileSchedule>
                <fileURI>file://fota.operator.com/swupdate/oem-1/model-2/image098798.apk</fileURI>
                <deliveryInfo start="2012-03-01T23:10:00Z" end="2012-03-01T23:20:00Z"/>
                <sv:delimiter>0</sv:delimiter>
            </fileSchedule>
            <fileSchedule>
                <fileURI>file://fota.operator.com/swupdate/oem-1/model-3/image765987.apk</fileURI>
                <deliveryInfo start="2012-03-01T23:20:00Z" end="2012-03-01T23:30:00Z"/>
                <sv:delimiter>0</sv:delimiter>
```

```
            </fileSchedule>
        </serviceSchedule>
    <serviceSchedule>
         <sessionSchedule>
            <start>2012-03-07T10:00:00Z</start>
             <stop>2012-03-07T10:30:00Z</stop>
                <sv:delimiter>0</sv:delimiter>
         </sessionSchedule>
         <fileSchedule>
             <fileURI>file://fota.operator.com/swupdate/oem-1/model-4/image456345.apk</fileURI>
             <deliveryInfo start="2012-03-07T10:00:00Z" end="2012-03-07T10:15:00Z"/>
                <sv:delimiter>0</sv:delimiter>
         </fileSchedule>
         <fileSchedule>
             <fileURI> file://fota.operator.com/swupdate/oem-1/model-5/image504123.apk</fileURI>
             <deliveryInfo start="2012-03-07T10:15:00Z" end="2012-03-07T10:30:00Z"/>
                <sv:delimiter>0</sv:delimiter>
         </fileSchedule>
     </serviceSchedule>
</scheduleDescription>
```

Example 2

In this example, the following key points can be noted:

- The version of the schema used to generate this instantiation is version 2, as indicated by the schemaVersion element. Thus a receiver of this instantiation that has multiple schedule schema versions (say 1, 2 and 3), then it should use a schedule schema version equals to 2, as indicated in the version attribute of the schema instruction "schema". A receiver that has only version 1 of the schedule schema version should use that schema for validating the instantiation, but extensions made in version 2 of the schema will be ignored by the receiver. The receiver should avoid using version 3 of the schema, as the instantiation would fail verification against that schema version.
- The scheduleUpdate element indicates UTC time February 1st at time 00 hour 00 min 00 sec. Thus UE should seek to update the schedule instance after this time.
- There is 1 serviceSchedule element, which includes:
    - a sessionSchedule that
        - Has a first occurrence that starts at UTC March 7th 2012 23:00:00 and ends at UTC March 7th 2012 23:30:00;
        - Has subsequent occurrences everyday at the same time until UTC March 14th 2012 00:00:00
    - A first fileSchedule used for FOTA for oem-1 model-1 with a filename of image032212.apk, which delivery starts at UTC March 7th 2012 23:00:00 and ends at March 7th 2012 23:10:00, thus 10 minutes duration;
    - A second fileSchedule used for FOTA for oem-1 model-2 with a filename of image098798.apk, which delivery starts at UTC March 7th 2012 23:10:00 and ends at March 7th 2012 23:20:00, thus 10 minutes duration;
    - A third fileSchedule used for FOTA for oem-1 model-3 with a filename of image765987.apk, which delivery starts at UTC March 7th 2012 23:20:00 and ends at March 7th 2012 23:30:00, thus 10 minutes duration;

```
<?xml version="1.0" encoding="UTF-8"?>
<scheduleDescription
    xmlns="urn:3gpp:metadata:2011:MBMS:scheduleDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:3gpp:metadata:2011:MBMS:scheduleDescription Schedule-Description-
Main.xsd"
    scheduleUpdate="2012-02-07T00:00:00Z"
```

```
    version="764">
    <sv:schemaVersion>2</sv:schemaVersion>
    <serviceSchedule>
        <sessionSchedule>
            <start>2012-03-07T23:00:00Z</start>
            <stop>2012-03-07T23:30:00Z</stop>
            <reoccurencePattern>daily</reoccurencePattern>
            <reoccurenceStopTime>2012-03-14T00:00:00Z</reoccurenceStopTime>
                <sv:delimiter>0</sv:delimiter>
        </sessionSchedule>
        <fileSchedule>
            <fileURI>file://fota.operator.com/swupdate/oem-1/model-1/image032212.apk</fileURI>
            <deliveryInfo start="2012-03-07T23:00:00Z" end="2012-03-07T23:10:00Z"/>
                <sv:delimiter>0</sv:delimiter>
        </fileSchedule>
        <fileSchedule>
            <fileURI>file://fota.operator.com/swupdate/oem-1/model-2/image098798.apk</fileURI>
            <deliveryInfo start="2012-03-07T23:10:00Z" end="2012-03-07T23:20:00Z"/>
                <sv:delimiter>0</sv:delimiter>
        </fileSchedule>
        <fileSchedule>
            <fileURI>file://fota.operator.com/swupdate/oem-1/model-3/image765987.apk</fileURI>
            <deliveryInfo start="2012-03-07T23:20:00Z" end="2012-03-07T23:30:00Z"/>
                <sv:delimiter>0</sv:delimiter>
        </fileSchedule>
    </serviceSchedule>
</scheduleDescription>
```

Example 3

In this example, the following key points can be noted:

- The version of the schema used to generate this instantiation is version 2, as indicated by the schemaVersion element. Thus a receiver of this instantiation that has multiple schedule schema versions (say 1, 2 and 3), then it should use a schedule schema version equals to 2, as indicated in the version attribute of the schema instruction “schema”. A receiver that has only version 1 of the schedule schema version should use that schema for validating the instantiation, but extensions made in version 2 of the schema will be ignored by the receiver. The receiver should avoid using version 3 of the schema, as the instantiation would fail verification against that schema version.
- The scheduleUpdate element indicates UTC time February 1st at time 00 hour 00 min 00 sec. Thus UE should seek to update the schedule instance after this time.
- There is 1 serviceSchedule element, which includes:
    - a sessionSchedule that
        - Has a first occurrence that starts at UTC March 1st 2012 23:00:00 and ends at UTC March 1st 2012 23:30:00;
        - Has subsequent occurrences everyday at the same time until UTC March 7th 2012 00:00:00

```
<?xml version="1.0" encoding="UTF-8"?>
<scheduleDescription xmlns="urn:3gpp:metadata:2011:MBMS:scheduleDescription"
    scheduleUpdate="2012-02-01T00:00:00Z"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    xsi:schemaLocation="urn:3gpp:metadata:2011:MBMS:scheduleDescription Schedule-Description-
Main.xsd">
    <sv:schemaVersion>2</sv:schemaVersion>
    <serviceSchedule>
        <sessionSchedule>
            <start>2012-03-01T23:00:00Z</start>
            <stop>2012-03-01T23:30:00Z</stop>
            <reoccurencePattern>daily</reoccurencePattern>
            <reoccurenceStopTime>2012-03-07T00:00:00Z</reoccurenceStopTime>
                <sv:delimiter>0</sv:delimiter>
        </sessionSchedule>
```

```
    </serviceSchedule>
</scheduleDescription>
```

11.2B Filter Description Metadata Fragment

11.2B.1 Introduction

This clause describes the procedures, usage and XML schema of the Filter Description metadata fragment. In this version of the specification, the Filter Description fragment, if present, shall only contain location filtering data. The procedures associated with the elements of the Filter Description metadata fragment are specified in sub-clause 11.2B.2. The basic buildng block of a location filter is referred to as the location rule. Multiple location rules or constituent location filters built from these rules can be combined by using logical relationships (AND, OR, NOT) and recursion to form the overall, composite location filter. Such usage and semantics are described in sub-clause 11.2B.3. The XML schema for the Filter Description Metadata Fragment is specified in sub-clause 11.2B.4.

The UE may support the Filter Description fragment, and it may support the location filter data carried in this fragment.

11.2B.2 Procedures for Filter Description Metadata Fragment

A Filter Description metadata fragment instance may be delivered to the MBMS clients:

- during MBMS User Service Discovery/Announcement prior to the MBMS Download delivery session along with the session description (out-of-band of that session); or
- in-band within a MBMS Download delivery session; or
- via an MBMS download delivery session dedicated to the transport of filter metadata information.

The most recently delivered Filter Description instance (i.e. the one with the highest version number - as given in the MBMS Metadata Envelope, see sub-clause 11.1.3) shall take the highest priority, such that filter data parameters received prior to, and out-of-band, of the download session they apply to are regarded as "initial defaults", and filter data parameters received in-band with the download session overwrite the earlier received filter data parameters.

The Filter Description instance is clearly identified using a URI to enable UE cross-referencing of in and out-of-band filter data files.

The MIME Type for the filter data information is "application/mbms-filter+xml"

Availability of the Filter Description fragment is indicated by the presence of the *receptionFiltering* element in the Schedule Description fragment, as child element of either the *sessionSchedule* or the *fileSchedule* element. Each *filterData* element is identified by its *id* attribute, which enables cross-referencing with the *filterID* attribute in the Schedule Description fragment.

11.2B.3 Usage of Location Filter

The overall location filter, representing the filtering criteria for positive reception of an associated location-specific content item or service, is expressed by the contents of the complex element *locationFilter* whose syntax is specified by the schema of the Filter Description fragment in clause 11.2B.4. The fundamental component or “atomic” building block of the location filter is the child element *locationRule*. It may be combined, via the element *logicalOperation* along with the use of recursion, with additional location rules and/or intermediate location filters to form the composite location filter. From the schema in 11.2B.4, possible decompositions of the location filter are as follows:

a) LF = *locationFilter* = *locationFilter1* = LF1;

b) “ = *locationFilter2* = LF2;

c) “ = *locationRule* = LR;

d) “ = *locationFilter1* *logicalOperation* *locationRule* = LF1 **LO** LR;

e) “ = *locationFilter1* *logicalOperation* *locationFilter2* = LF1 **LO** LF2;

f) " = *logicalOperation locationFilter2* = **LO** LF2;

The string value of *logicalOperation* (abbreviated as **LO** in the above) may correspond to the binary operator 'AND' or 'OR', or the unary operator 'NOT'. In the decomposition of (f), the only allowed value for this element is 'NOT'.

The basic location rule, defined by the element *locationRule*, comprises one or more target areas for which the required presence may additionally be associated with temporal and confidence level criteria. Each target area, represented by the child element *targetArea*, shall in turn comprise one or more cell-IDs and/or one or more shapes, the latter parameter defined by OMA MLP [109]. Permitted shapes in this version of the specification are 'Polygon' and 'CircularArea' as defined by [109]. If present, temporal parameters shall include the elements *startTime* and *endTime*, and optionally the element *duration*. Presence of only *startTime* and *endTime* is an indication that reception of the associated content item or service is conditioned by the location of the UE in any one of the instantiated target areas for the entire interval given by the difference between the start and end time values. Additional presence of *duration* means that UE presence in the target area is only required for the interval given by the value of this element, with *duration* < (*endTime – startTime*). In addition to temporal criteria, targeted reception of the content or service may be further conditioned by a minimum required level of confidence from the UE's perspective, and given by the element *confidenceLevel*, that the location of the UE fulfills both the associated target area and time criteria. In the event that the filter criterion corresponds to the present/current UE location in a target area defined strictly by cell-ID(s), both the time and confidence level parameters shall be absent. Defining targeted reception of a content or service by multiple time intervals requires multiple instances of *locationRule* in the composite location filter, since a single location rule can only be associated with a single temporal criterion.

The evaluation of whether an instance of location filtering criteria is satisfied is a device-internal mechanism and implementation-specific issue outside the scope of this specification.

11.2B.4 XML Schema for the Filter Description Metadata Fragment

Indicated below is the formal XML syntax of the Filter Description metadata fragment, containing location filtering data as represented by the element *locationFilter*. Documents following this schema can be identified with the MIME type "application/mbms-filter+xml" defined in Annex C.15.

In this version of the specification the network shall set the *schemaVersion* element, defined as a child of *filterDescription* 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 a UE receives an instantiation of a Filter Description compliant to this schema, it shall determine the schema version required to parse the instantiation as follows:

- If the UE supports one or more versions of the Filter Description schema with the schema *version* attribute, then the UE 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 schema filename of the Filter Description schema is FilterDescription.xsd. The XML schema "schema-version.xsd" is specified in Annex J.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3gpp:metadata:2011:MBMS:filterDescription"
     xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    targetNamespace="urn:3gpp:metadata:2011:MBMS:filterDescription" elementFormDefault="qualified"
    version="1">
<xs:import schemaLocation="schema-version.xsd"
namespace="urn:3gpp:metadata:2009:MBMS:schemaVersion"/>
    <xs:complexType name="locationFilterType">
        <xs:sequence>
         <xs:element ref="sv:schemaVersion"/>
            <xs:element name="locationFilter1" type="locationFilterType" minOccurs="0"/>
            <xs:element name="logicalOperation" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="AND"/>
                        <xs:enumeration value="OR"/>
```

```
                        <xs:enumeration value="NOT"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:choice>
                <xs:element name="locationRule" minOccurs="0">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="targetArea" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="cellID" type="xs:unsignedLong"
                                      minOccurs="0" maxOccurs="unbounded"/>
                                        <xs:element name="shape" minOccurs="0">
                                            <xs:complexType>
                                                <xs:sequence>
                                                  <xs:element name="Polygon" minOccurs="0"
                                                   maxOccurs="unbounded">
                                                        <xs:annotation>
                                                            <xs:documentation>see [OMA
MLP]</xs:documentation>
                                                        </xs:annotation>
                                                   </xs:element>
                                                  <xs:element name="CircularArea" minOccurs="0"
                                                        maxOccurs="unbounded">
                                                        <xs:annotation>
                                                            <xs:documentation>see [OMA
MLP]</xs:documentation>
                                                        </xs:annotation>
                                                   </xs:element>
                                                   <xs:any namespace="##other"
processContents="lax"
                                                    minOccurs="0" maxOccurs="unbounded"/>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:any namespace="##other" processContents="lax"
                                        minOccurs="0" maxOccurs="unbounded"/>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="startTime" type="xs:unsignedInt" minOccurs="0"/>
                            <xs:element name="endTime" type="xs:unsignedInt" minOccurs="0"/>
                            <xs:element name="duration" type="xs:unsignedInt" minOccurs="0"/>
                            <xs:element name="confidenceLevel" minOccurs="0">
                                <xs:simpleType>
                                    <xs:restriction base="xs:decimal">
                                        <xs:minInclusive value="0"/>
                                        <xs:maxInclusive value="1"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="locationFilter2" type="locationFilterType" minOccurs="0"/>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="locationRuleType">
        <xs:sequence>
            <xs:element name="targetArea" type="targetAreaType" maxOccurs="unbounded"/>
            <xs:element name="startTime" type="xs:unsignedInt" minOccurs="0"/>
            <xs:element name="endTime" type="xs:unsignedInt" minOccurs="0"/>
            <xs:element name="duration" type="xs:unsignedInt" minOccurs="0"/>
            <xs:element name="confidenceLevel" minOccurs="0">
                <xs:simpleType>
                    <xs:restriction base="xs:decimal">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="targetAreaType">
        <xs:sequence>
            <xs:element name="cellID" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>
```

```
            <xs:element name="shape">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Polygon" minOccurs="0" maxOccurs="unbounded">
                            <xs:annotation>
                                <xs:documentation>see [OMA MLP]</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                        <xs:element name="CircularArea" minOccurs="0" maxOccurs="unbounded">
                            <xs:annotation>
                                <xs:documentation>see [OMA MLP]</xs:documentation>
                            </xs:annotation>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="filterDescriptionType">
        <xs:sequence>
            <xs:element name="filterData" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="locationFilter" type="locationFilterType" minOccurs="0"/>
                        <xs:any namespace="##other" processContents="lax" minOccurs="0"
                         maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="id" type="xs:anyURI" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:element name="filterDescription" type="filterDescriptionType"/>
</xs:schema>
```

11.3 Service Protection Description Metadata Fragment

11.3.1 Definition of the Protection Description

The security description is referenced by the *protectionDescriptionURI* of a deliveryMethod element. The security description fragment shall be identified by the MIME type "application/mbms protection-description+xml" as defined in Annex C.6.

The root element of the security description is the *securityDescription* element. It contains three different elements, the *keyId* element identities the key(s) for each media flow, the *keyManagement* element the keymanagement servers that the load is distributed over and the parameters in use, and the *fecProtection* element that configures the FEC protection of the point to multi-point distributed key flows.

The *keyManagement* element defines the list of key management servers (i.e. BM-SC). The MBMS UE must register with a key management server to receive key material. A receiver shall select a key managment server following the same procedure defined for selecting a file repair server defined in sub-clause 9.3.5.

The attribute *uiccKeyManagement* defines whether UICC based key management is required for the present MBMS User Service.

The *offsetTime* and *randomTimePeriod* attributes define the back off behavior of the UE when requesting MSKs. This uses the procedure defined in sub-clause 9.3.4 where *offsetTime* specifies the offset time defined in sub-clause 9.3.4.1 and *randomTimePeriod* the length of the random window in accordance with sub-clause 9.3.4.2. The units for both attributes are in seconds.

The element *keyId* contains a list of media flows for which keys are required. For each media flow a key identifier is provided in addition to that media flows additional security parameters. The media flow is identified by a destination tuple providing an address followed by a port number separated by a "/", i.e. <IP-destination-address>/<destination-port>. The port number is for RTP session the RTP port number, and not RTCP's. The *MSK* element identifies the key uniquely by specifying both the *keyDomainID* and the *MSKID* as defined in sub-clause 6.3.2.1 of 3GPP TS 33.246 [20]. The *MSKID* is 4 bytes long binary with byte 3 and 4 equal to 0x00, i.e. the current key that are base64 [82] when written into the element. The *keyDomainID* is a 3 byte long binary value as specified in sub-clause 6.3.2.1 of [20] and shall also be base64 encoded when written in the XML document.

The presence of the *fecProtection* element indicates that any MIKEY packet with an multicast destination IP address equal to any of the used destination address in the User Service Bundle Description instance's delivery methods, are FEC protected and encapsulated in FEC source packets, see sub-clause 8.2.2.4. The attributes *fecEncodingId*, *fecInstanceID*, and *fecOtiExtension* specify the FEC payload ID used in the source packet. All service protection descriptions referenced by a User Service Bundle Description instance shall use the same FEC parameters.

The schema filename of Security Description (as defined below) is "security.xsd":

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2005:MBMS:securityDescription"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3GPP:metadata:2005:MBMS:securityDescription" elementFormDefault="qualified">
    <xs:element name="securityDescription" type="securityDescriptionType"/>
    <xs:complexType name="securityDescriptionType">
        <xs:sequence>
            <xs:element name="keyManagement" type="keyManagementType" minOccurs="0"/>
            <xs:element name="keyId" type="keyIdType" maxOccurs="unbounded"/>
            <xs:element name="fecProtection" type="fecProtectionType" minOccurs="0"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="keyManagementType">
        <xs:sequence>
            <xs:element name="serverURI" type="xs:anyURI" maxOccurs="unbounded"/>
            <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
        </xs:sequence>
        <xs:attribute name="offsetTime" type="xs:unsignedLong" use="optional" default="0"/>
        <xs:attribute name="randomTimePeriod" type="xs:unsignedLong" use="optional" default="0"/>
        <xs:attribute name="uiccKeyManagement" type="xs:boolean" use="optional" default="true"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="keyIdType">
        <xs:sequence>
            <xs:element name="mediaFlow" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="MSK" type="MSKType" maxOccurs="1"/>
                    </xs:sequence>
                    <xs:attribute name="flowID" type="xs:string" use="required"/>
                    <xs:anyAttribute processContents="skip"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="fecProtectionType">
        <xs:attribute name="fecEncodingId" type="xs:unsignedLong" use="optional" default="0"/>
        <xs:attribute name="fecInstanceId" type="xs:unsignedLong" use="optional"/>
        <xs:attribute name="fecOtiExtension" type="xs:string" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="MSKType">
        <xs:sequence>
            <xs:element name="keyDomainID" type="xs:base64Binary" minOccurs="1" maxOccurs="1"/>
            <xs:element name="MSKID" type="MSKIDType" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="MSKIDType">
        <xs:restriction base="xs:base64Binary">
            <xs:length value="4"/>
        </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

11.3.2 Example of a Security Description:

```
<?xml version="1.0" encoding="UTF-8"?>
<securityDescription
    xmlns="urn:3GPP:metadata:2005:MBMS:securityDescription"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:securityDescription security.xsd">
    <keyManagement
        offsetTime="5"
        randomTimePeriod="10"
        uiccKeyManagement="true">
        <serverURI>http://register.operator.umts/</serverURI>
        <serverURI>http://register2.operator.umts/</serverURI>
    </keyManagement>
    <keyId>
        <mediaFlow flowID="224.1.2.3/4002">
            <MSK>
                <keyDomainID>aMoM</keyDomainID>
                <MSKID>aMoAAA==</MSKID>
            </MSK>
        </mediaFlow>
        <mediaFlow flowID="224.1.2.3/4004">
            <MSK>
                <keyDomainID>GM8M</keyDomainID>
                <MSKID>aMkAAA==</MSKID>
            </MSK>
        </mediaFlow>
    </keyId>
    <fecProtection
        fecEncodingId="1"
        fecInstanceId="0"
        fecOtiExtension="1SCxWEMNe397m24SwgyRhg==">/>
</securityDescription>
```

11.4 Service Protection Registration Format

11.4.1 Data Format

The below XML schema defines a format used to register to the keymanagement servers according to the procedure in TS 33.246. The MIME type for this format is defined in appendix C.9. The *serviceID* element identifies the service uniquely and is the same *serviceID* used in the userServiceDescription format defined in sub-clause 11.2.1. The schema filename of service protection registration is SecurityRegistration.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3GPP:metadata:2005:MBMS:securityRegistration"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="mbmsSecurityRegister">
        <xs:annotation>
            <xs:documentation>MBMS Security Registration according to TS 33.246</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="serviceID" type="xs:anyURI" maxOccurs="unbounded" minOccurs="1"/>
                <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
                    processContents="lax"/>
            </xs:sequence>
        <xs:anyAttribute processContents="skip"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

11.4.2 Example

The below example is used to register to a service identifed by the serviceID " urn:3gpp:mbms:example:serivce:identification:123456789abcdef".

```
<?xml version="1.0" encoding="UTF-8"?>
<mbmsSecurityRegister xmlns="urn:3GPP:metadata:2005:MBMS:securityRegistration"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:securityRegistration SecurityRegistration.xsd">
    <serviceID>urn:3gpp:mbms:example:serivce:identification:123456789abcdef</serviceID>
```

```
</mbmsSecurityRegister>
```

11.5 Service Protection De-Registration Format

11.5.1 Data Format

This format is used to de-register from the keymanagement server(s) according to the procedure in TS 33.246. The MIME type for this format is defined in appendix C.10. The *serviceID* element is defined exactly as in sub-clause 11.4.1. The schema filename of service protection de-registration is SecurityDeregistration.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:3GPP:metadata:2005:MBMS:securityDeregistration"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="mbmsSecurityDeregister">
        <xs:annotation>
            <xs:documentation>MBMS Security Deregistration according to TS 33.246</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="serviceID" type="xs:anyURI" maxOccurs="unbounded" minOccurs="1"/>
                <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
                    processContents="lax"/>
            </xs:sequence>
        <xs:anyAttribute processContents="skip"/>
        </xs:complexType>
        </xs:element>
</xs:schema>
```

11.5.2 Example

The below example is used to de-register from the services identifed by the serviceID " urn:3gpp:mbms:example:serivce:identification:123456789abcdef".

```
<?xml version="1.0" encoding="UTF-8"?>
<mbmsSecurityDeregister xmlns="urn:3GPP:metadata:2005:MBMS:securityDeregistration"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
The schema filename of service protection de-registration is SecurityDeregistration.xsd.
    <serviceID>urn:3gpp:mbms:example:serivce:identification:123456789abcdef</serviceID>
</mbmsSecurityDeregister>
```

11.6 Service Protection MSK Request Format

11.6.1 Data Format

This format is used to request from the keymanagement server(s) the delivery of one or more MSK identities as defined in sub-clause 11.3.1. The MIME type for this format is defined in appendix C.8. The schema filename of service protection de-registration is SecurityDeregistration.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:3GPP:metadata:2005:MBMS:mskRequest"
    targetNamespace="urn:3GPP:metadata:2005:MBMS:mskRequest"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

    <xs:element name="mbmsMSKRequest">
        <xs:annotation>
            <xs:documentation>
                MBMS MSK Request as defined by 3GPP TS 26.346 and 3GPP TS 33.246
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="MSK" type="MSKType" minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:anyAttribute processContents="skip"/>
        </xs:complexType>
    </xs:element>
```

```
    <xs:complexType name="MSKType">
        <xs:sequence>
            <xs:element name="keyDomainID" type="xs:base64Binary" minOccurs="1" maxOccurs="1"/>
            <xs:element name="MSKID" type="MSKIDType" minOccurs="1" maxOccurs="1"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="MSKIDType">
            <xs:restriction base="xs:base64Binary">
                <xs:length value="4"/>
            </xs:restriction>
    </xs:simpleType>
</xs:schema>
```

11.6.2 Example

The below example is used to request a single MSK with keyDomainID "uHCd" and a MSK ID part "aMkAAA==".

```
<?xml version="1.0" encoding="UTF-8"?>
<mbmsMSKRequest
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="urn:3GPP:metadata:2005:MBMS:mskRequest">
xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:mskRequest mskRequest.xsd">
    <MSK>
        <keyDomainID>uHCd</keyDomainID>
        <MSKID>aMkAAA==</MSKID>
    </MSK>
</mbmsMSKRequest>
```

11.7 Service Protection Registration and De-Registration Response Format

11.7.1 Data Format

This format is used in the response of the keymanagement server(s) to a Service Protection Registration or De-Registration message. Service Protection Registration message format is defined in clause 11.4 and the Service Protection De-Registration message format in clause 11.5. The format of the response codes are defined in 3GPP TS 33.246 [20]. The MIME Media type for this format is defined in appendix C.13. The schema filename of service protection registration and de-registration response format is securityRegistrationResponse.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2005:MBMS:securityRegistrationResponse"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3GPP:metadata:2005:MBMS:securityRegistrationResponse"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="mbmsSecurityRegisterResponse">
        <xs:annotation>
            <xs:documentation>MBMS Security Registration Response according to TS
33.246</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Response" type="ResponseType" maxOccurs="unbounded"/>
                <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="ResponseType">
        <xs:sequence>
            <xs:element name="serviceID" type="xs:anyURI"/>
            <xs:element name="ResponseCode" type="xs:string"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

11.7.2 Example

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

```
<mbmsSecurityRegisterResponse xmlns="urn:3GPP:metadata:2005:MBMS:securityRegistrationResponse"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:securityRegistrationResponse">
securityRegistrationResponse.xsd">
    <Response>
        <serviceID>urn:3gpp:mbms:example:service:identification:123456789abcdef</serviceID>
        <ResponseCode>200 OK</ResponseCode>
    </Response>
    <Response>
        <serviceID>urn:3gpp:mbms:example:service:identification:fedcba987654321</serviceID>
        <ResponseCode>200 OK</ResponseCode>
    </Response>
</mbmsSecurityRegisterResponse>
```

11.8 Service Protection MSK Response Format

11.8.1 Data Format

This format is used in the response of the keymanagement server(s) to an MSK Request message. The MSK Request message format is defined in clause 11.6. The format of the response codes are defined in 3GPP TS 33.246 [20]. The MIME Media type for this format is defined in appendix C.12. The schema filename of service protection MSK response format is mskResponse.xsd.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="urn:3GPP:metadata:2005:MBMS:mskResponse"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:3GPP:metadata:2005:MBMS:mskResponse" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:element name="mbmsMSKResponse">
        <xs:annotation>
            <xs:documentation>MBMS Security MSK Request Response according to TS
33.246</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Response" type="ResponseType" maxOccurs="unbounded"/>
                <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="MSKType">
        <xs:sequence>
            <xs:element name="keyDomainID" type="xs:base64Binary"/>
            <xs:element name="MSKID" type="MSKIDType"/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="MSKIDType">
        <xs:restriction base="xs:base64Binary">
            <xs:length value="4"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="ResponseType">
        <xs:sequence>
            <xs:element name="MSK" type="MSKType"/>
            <xs:element name="ResponseCode" type="xs:string"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

11.8.2 Example

```
<?xml version="1.0" encoding="UTF-8"?>
<mbmsMSKResponse xmlns="urn:3GPP:metadata:2005:MBMS:mskResponse"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:mskResponse mskResponse.xsd">
    <Response>
        <MSK>
            <keyDomainID>uHCd</keyDomainID>
            <MSKID>aMkAAA==</MSKID>
```

```
        </MSK>
        <ResponseCode>200 OK</ResponseCode>
    </Response>
    <Response>
        <MSK>
            <keyDomainID>uHCd</keyDomainID>
            <MSKID>JMtEAA==</MSKID>
        </MSK>
        <ResponseCode>200 OK</ResponseCode>
    </Response>
</mbmsMSKResponse>
```

11.9 MBMS Feature Requirements

MBMS features enable the BM-SC to signal to the UE the set of capabilities that are required for the consumption of the MBMS user service. The required capability list is indicated in the MBMS User Service Description of the corresponding MBMS user service as defined in section 11.2.1.

The MBMS UE shall not attempt to receive the service if it detects that at least one required capability, indicated in the USD, is not supported or not understood. The introduction of new features is possible and assumes that unidentified features shall be interpreted by the UE as a requirement that cannot be fulfilled.

The following list of features is currently identified:

Table 2 - MBMS Feature Requirement List

Service Capability	References	Recognized Feature Values (Integer)
Speech	as defined in clause 10.2	0
AMR-WB	as defined in clause 10.2	1
Enhanced aacPlus	as defined in clause 10.3	2
Extended AMR-WB	as defined in clause 10.3	3
Synthetic audio	as defined in clause 10.4	4
H.263	as mentioned in clause 10.5	5
H.264 Constrained Baseline Profile Level 1b	as defined in clause 10.5 (of Release 6)	6
Still images	as defined in clause 10.6	7
Bitmap graphics	as defined in clause 10.7	8
Vector graphics	as defined in clause 10.8	9
Text	as defined in clause 10.9	10
Timed text	as defined in clause 10.10	11
3GPP file format	as defined in clause 10.11	12
H.264 Constrained Baseline Profile Level 1.2	as defined in clause 10.5 (of Release 7)	13
Scene Description	as defined in clause 10.12	14
MBSFN mode in UTRAN	as defined in 3GPP TS 25.346 (of Release 7)	15
H.264 Constrained Baseline Profile Level 1.3	as defined in clause 10.5 (of Release 9)	16
AHS	as defined in clause 5.6 (of Release 9)	17
3GP-DASH	as defined in clause 5.6 (of Release 10)	18
H.264 Progressive High Profile Level 3.1	as defined in clause 10.5 (of Release 11)	19
Frame-packed stereoscopic 3D video	as defined in clause 10.5 (of Release 11)	20

The list of features may be extended in the future.

Annex A (normative): FLUTE Support Requirements

This clause provides a table representation of the requirement levels for different features in FLUTE. Table A.1 includes requirements for an MBMS client and an MBMS server for FLUTE support as well as the requirements for a FLUTE client and a FLUTE server according to the FLUTE protocol (RFC 3926 [9]). The terms used in table A.1 are described underneath.

Table A.1: Overview of the FLUTE support requirements in MBMS servers and clients

	FLUTE Client support requirement as per [9]	**MBMS FLUTE Client support requirement as per present document**	**FLUTE Server use requirement as per [9]**	**MBMS FLUTE Server use requirement as per present document**
FLUTE Blocking Algorithm	Required	Required	Strongly recommended	Required
Symbol Encoding Algorithm	Compact No-Code algorithm required. Other FEC building blocks are undefined optional plug-ins.	Compact No-Code algorithm required. MBMS Forward Error Correction required	Compact No-Code algorithm is the default option. Other FEC building blocks are undefined optional plug-ins.	Compact No-Code algorithm is the default option. MBMS Forward Error Correction.
Congestion Control Building Block (CCBB) / Algorithm	Congestion Control building blocks undefined.	Single channel support required	Single channel without additional CCBB given for the controlled network scenario.	Single channel support required
Content Encoding for FDT Instances	Optional	Optional	Optional	Shall not be used
Content Encoding for any other file than FDT Instances	Optional	Required	Optional	Optional
A flag active (header)	Required	Required	Optional	Not recommended to use
B flag active (header)	Required	Required	Optional	Not recommended to use
T flag active and SCT field (header)	Optional	Optional	Optional	Set to zero
R flag active and ERT field (header)	Optional	Optional	Optional	Set to zero
Content-Location attribute (FDT)	Required	Required	Required	Required
TOI (FDT)	Required	Required	Required	Required
FDT Expires attribute (FDT)	Required	Required	Required	Required
Complete attribute (FDT)	Required	Required	Optional	Optional
FEC-OTI-Maximum-Source-Block-Length	Required	Required	Required	Required
FEC-OTI-Encoding-Symbol-Length	Required	Required	Required	Required
FEC-OTI-Max-Number-of-Encoding-Symbols.	Required	Required	Required	Required
FEC-OTI-FEC-Instance-ID	Required	Optional	Required	Optional
FEC-OTI-Scheme-Specific-Info	n/a	Required	n/a	Required if MBMS FEC used

The following are descriptions of the above terms:

- **Blocking algorithm:** The blocking algorithms is used for the fragmentation of files. It calculates the source blocks from the source files.
- **Symbol Encoding algorithm:** The symbol encoding algorithm is used for the fragmentation of files. It calculates encoding symbols from source blocks for Compact No-Code FEC. It may also be used for other FEC schemes.
- **Congestion Control Building Block:** A building block used to limit congestion by using congestion feedback, rate regulation and receiver controls (RFC 3048 [17]).
- **Content Encoding for FDT Instances:** FDT Instance may be content encoded for more efficient transport, e.g. using GZIP.
- **Content Encoding for any other file than FDT Instances:** Files may be content encoded for more efficient transport, e.g. using GZIP.
- **A flag:** The Close Session flag for indicating the end of a session to the receiver in the ALC/LCT header.
- **B flag:** The Close Object flag is for indicating the end of an object to the receiver in the ALC/LCT header.
- **T flag:** The T flag is used to indicate the use of the optional "Sender Current Time (SCT)" field (when T=1) in the ALC/LCT header.
- **R flag:** The R flag is used to indicate the use of the optional "Expected Residual Time (ERT) field in the ALC/LCT header.
- **Content Location attribute:** This attribute provides a URI for the location where a certain piece of content (or file) being transmitted in a FLUTE session is located.
- **Transport Object Identifier (TOI):** The TOI uniquely identifies the object within the session from which the data in the packet was generated.
- **FDT Expires attribute:** Indicates to the receiver the time until which the information in the FDT is valid.
- **Complete attribute:** This may be used to signal that the given FDT Instance is the last FDT Instance to be expected on this file delivery session.
- **FEC-OTI-Maximum-Source-Block-Length:** This parameter indicates the maximum number of source symbols per source block.
- **FEC-OTI-Encoding-Symbol-Length:** This parameter indicates the length of the Encoding Symbol in bytes.
- **FEC-OTI-Max-Number-of-Encoding-Symbols:** This parameter indicates the maximum number of Encoding Symbols that can be generated for a source block.
- **FEC-OTI-FEC-Instance-ID:** This field is used to indicate the FEC Instance ID, if a FEC scheme is used.
- FEC-OTI-Scheme-Specific-Info: Carries Object Transmission Information which is specific to the FEC scheme in use.

Annex B (normative): FEC encoder specification

This Annex specifies the systematic Raptor forward error correction code and its application to MBMS [7]. Raptor is a fountain code, i.e., as many encoding symbols as needed can be generated by the encoder on-the-fly from the source symbols of a block. The decoder is able to recover the source block from any set of encoding symbols only slightly more in number than the number of source symbols.

The code described in this document is a Systematic code, that is, the original source symbols are sent unmodified from sender to receiver, as well as a number of repair symbols.

B.1 Void

B.2 Void

B.3 File download

B.3.1 Void

B.3.2 Void

B.3.3 Void

B.3.4 Example parameters

B.3.4.1 Parameter derivation algorithm

Section 4.2 of [91] provides recommendations for the derivation of the transport parameters G, T, Z and N.

Recommended settings for the input parameters, W, Al, K_{MIN} and G_{MAX} are as follows:

W = 256 KB Al = 4 K_{MIN} = 1024 G_{MAX} = 10

B.3.4.2 Examples

The above algorithm leads to transport parameters as shown in Table B.3.4.2-1 below, assuming the recommended values for W, Al, K_{MIN} and G_{MAX} and P = 512:

Table B.3.4.2-1

File size F	G	Symbol size T	$G*T$	K_t	Source blocks Z	Sub-blocks N	K_L	K_S	$T_L \cdot A$	$T_S \cdot A$

100 KB	6	84	504	1,220	1	1	1,220	1,220	N/A	N/A
100 KB	8	64	512	1,600	1	1	1,600	1,600	N/A	N/A
300 KB	2	256	512	1,200	1	2	1,200	1,200	128	128
1,000 KB	1	512	512	2,000	1	5	2,000	2,000	104	100
3,000 KB	1	512	512	6,000	1	12	6,000	6,000	44	40
10,000 KB	1	512	512	20,000	3	14	6,666	6,667	40	36

B.4 Streaming

B.4.1 Void

B.4.2 Void

B.4.3 Void

B.4.4 Example parameters

B.4.4.1 Parameter derivation algorithm

This sub-clause provides recommendations for the derivation of the transport parameter *T*. This recommendation is based on the following input parameters:

- B the maximum source block size, in bytes
- P the maximum repair packet payload size, in bytes, which is a multiple of Al
- Al the symbol alignment factor, in bytes
- K_{MAX} the maximum number of source symbols per source block.
- K_{MIN} a minimum target on the number of symbols per source block
- G_{MAX} a maximum target number of symbols per repair packet

A requirement on these inputs is that $\text{ceil}(B/P) \leq K_{MAX}$. Based on the above inputs, the transport parameter *T* is calculated as follows:

Let,

$G = \min\{\text{ceil}(P \cdot K_{MIN}/B), P/Al, G_{MAX}\}$ - the approximate number of symbols per packet

$T = \text{floor}(P/(Al \cdot G)) \cdot Al$

The value of *T* derived above should be considered as a guide to the actual value of *T* used. It may be advantageous to ensure that *T* divides into *P*, or it may be advantageous to set the value of *T* smaller to minimize wastage when full size

repair symbols are used to recover partial source symbols at the end of lost source packets (as long as the maximum number of source symbols in a source block does not exceed K_{MAX}). Furthermore, the choice of T may depend on the source packet size distribution, e.g., if all source packets are the same size then it is advantageous to choose T so that the actual payload size of a repair packet P', where P' is a multiple of T, is equal to (or as few bytes as possible larger than) the number of bytes each source packet occupies in the source block.

Recommended settings for the input parameters, Al, K_{MIN} and G_{MAX} are as follows:

$Al = 4$ $K_{MIN} = 1024$ $G_{MAX} = 10$

B.4.4.2 Examples

The above algorithm leads to transport parameters as shown in Table B.4.4.2-1 below, assuming the recommended values for Al, K_{MIN} and G_{MAX} and $P = 512$:

Table B.4.4.2-1

Max source block size B	G	Symbol size T	$G \cdot T$
40 KB	10	48	480
160 KB	4	128	512
640 KB	1	512	512

B.5 Void

B.6 Void

B.7 Void

The two tables V_0 and V_1 are defined in [91], clause 5.6.1 and 5.6.2 respectively.

B.8 Void

Annex C (informative): IANA registration

This annex provides the required IANA registration.

C.1 Registration of SDP Protocol Identifiers for Source packet

This specification defines two new SDP protocol identificators for source packets. In compliance with the registration rules, these SDP protocol identificators are defined in [100].

Protocol identifier “UDP/MBMS-FEC/RTP/AVP” identifies a protocol combination of UDP[7], FEC source packets (see sub-clause 8.2.2.4), RTP [6] using the AVP profile [78]. This protocol identifier shall use the FMT space rules that are used for RTP/AVP.

Protocol identifier “UDP/MBMS-FEC/RTP/SAVP” identifies a protocol combination of UDP [7], FEC source packets (see sub-clause 8.2.2.4), and RTP [6] using the SAVP profile [77]. This protocol identifier shall use the FMT space rules that are used for RTP/AVP.

C.2 Registration of SDP Protocol identifier for repair packets

This specification defines one new SDP protocol identificator for FEC repair packets. In compliance with the registration rules, this SDP protocol identificator are defined in [100].

Protocol identifier “UDP/MBMS-REPAIR” identifies a protocol combination of UDP [7], FEC repair packets (see sub-clause 8.2.2.4). The FMT string is not used and shall be set to “*”.

C.3 Registration of MIME type "application/simpleSymbolContainer"

The MIME Type "application/simpleSymbolContainer" denotes that the message body is a simple container of encoding symbols for the file repair procedure (clause 9.3.5.2 - File Repair Response Message Format for HTTP Carriage of Repair Data).

Type name: application

Subtype name: simpleSymbolContainer

Required parameters:

None

Optional parameters:

None

Encoding considerations:

The content is binary and if transported in a protocol not capable of handling binary content then the object must be encoded, for example using base64 [82].

Security considerations:

The content of this media type are either source or repair symbols part of a binary file object. Thus it security considerations depends on the security requirements on the file object. As modification of the response message may corrupt the complete file object due to the FEC repair operations, integrity protection is recommended. Source authentication is also recommended to prevent man in the middle or spoofing attacks resulting in erronous repair symbols.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.4 Registration of MIME type "application/mbms-user-service-description+xml "

The MIME Type "application/mbms-user-service-description+xml" denotes that the message body is a user service description instance in accordance with the XML schema "urn:3GPP:metadata:2005:MBMS:userServiceDescription" (see sub-clause 11.2.1).

Type name: application

Subtype name: mbms-user-service-description+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

This media format is used to configure the receiver on how to participate in a service. This format is highly suspectible to manipulation or spoofing for attacks desring to misslead a receiver about a session. Both integrity protection and source authentication is recommended to prevent missleading of the receiver.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage: None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.5 Registration of MIME type "application/mbms-envelope+xml "

The MIME Type "application/mbms-envelope+xml" denotes that the message body is a metadata envelope according to the XML schema "urn:3gpp:metadata:2005:MBMS:envelope" (see sub-clause 11.1.3).

Type name: application

Subtype name: mbms-envelope+xml

Required parameters:

None

Optional parameters:

embedded: Declares if all objects described in this envelope instance are embedded within the this envelope instance. The value is either 0 (false) or 1 (true) and when this parameter is omitted the value defaults to 0 (false).

referenced: Declares if all objects described in this envelope instance are referenced, none is embedded within the envelope. The value is either 0 (false) or 1 (true) and when this parameter is omitted the value defaults to 0 (false).

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

This media format contains information about versioning and validity of files, telling the receiver if it should invalidate already received files or replace previously received files with new version contained or referenced within this format. Thus modifying or spoofing an envelope is a simple way of making denial of service attack. Thus source authentication and integrity protection of the files are recommended.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.6 Registration of MIME type "application/mbms-protection-description+xml "

The MIME-Type "application/mbms-protection-description+xml" denotes that the message body is an MBMS protection description XML document according to the scheme identified as "urn:3GPP:metadata:2005:MBMS:securityDescription" (see sub-clause 11.3.1).

Type name: application

Subtype name: mbms-protection-description+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The media format is used to configure the security functions of the MBMS delivery. Thus a target for any attacker that would like modify a receivers understanding of the security functions, either for a denial of service attack or a session replacement attack. However the format contains no seceret information that may not be shared openly. All security keys are handled using other mechanisms. Source authentication and integrity protection is recommended.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.7 Registration of MIME type "application/mbms-associated-procedure-description+xml"

The MIME-Type "application/mbms-associated-procedure-description+xml" denotes that the message body contains the associated procedure description in XML following the schema "urn:3gpp:metadata:2005:MBMS:associatedProcedure" (see sub-clause 9.5.1).

Type name: application

Subtype name: mbms-associated-procedure-description+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The information present in this media format is used to configure the receiving application. Thus the usage of the fomat is vulnerable to attacks modifying or spoofing the content of this format. It is recommended to use source authentication and integrity protection.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.8 Registration of MIME type "application/mbms-msk+xml"

The MIME-Type "application/ mbms-msk+xml" denotes that the message body contains the MSK request parameters in accordance with the XML scheme "urn:3GPP:metadata:2005:MBMS:mskRequest" (see sub-clause 11.6 and 3GPP TS 33.246 [20]).

Type name: application

Subtype name: application/mbms-msk+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The usage of the format is to identify one or more MSK that the requesting entity desires to receive. Modification or changes to this format thus only result in a denial of service attack. Integrity protection would protect against such modifications.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.9 Registration of MIME type "application/mbms-register+xml"

The MIME-Type "application/mbms-register+xml" denotes that the message body contains the MBMS User Service Registration parameters in accordance with the XML schema "urn:3GPP:metadata:2005:MBMS:securityRegistration" (see sub-clause 11.4 and 3GPP TS 33.246 [20]).

Type name: application

Subtype name: mbms-register+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The content of this format identifies a user service that the document creator desires to register to. Any modificatons of this document would allow an attacker to change to what services the creator registers to. To prevent this integrity protection is necessary. There is also necessary to authenticate the entity performing the registration to know who is the one performing the registration request. In MBMS this is accomplished using procedures described in 3GPP TS 33.246.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.10 Registration of MIME type "application/mbms-deregister+xml"

The MIME-Type "application/mbms-deregister+xml" denotes that the message body contains the MBMS User Service Deregistration parameters in accordance with the XML schema "urn:3GPP:metadata:2005:MBMS:securityDeregistration" (see sub-clause 11.5 of 3GPP TS 26.346 and 3GPP TS 33.246 [20]).

Type name: application

Subtype name: mbms-deregister+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The content of this format identifies a user service that the document creator desires to register to. Any modificatons of this document would allow an attacker to change to what services the creator registers to. To prevent this integrity protection is necessary. There is also necessary to authenticate the entity performing the registration to know who is the one performing the registration request. In MBMS this is accomplished using procedures described in 3GPP TS 33.246.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.11 Registration of MIME type "application/mbms-reception-report+xml"

The MIME-type registration for "application/mbms-reception-report+xml" that identifies XML documents that follows the schema "urn:3gpp:metadata:2005:MBMS:receptionreport" defined in clause 9.5.3.

Type name: application

Subtype name: mbms-reception-report+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The reception report XML document instances contain information about what services and files that a particular receiver has received. Thus to prevent manipulation of that information it would need to be integrity protected. The information also has privacy relevance as it reveals what a specific receiver, which usually can be connected to a specific user, has received.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

None

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.12 Registration of MIME type "application/mbms-msk-response+xml"

The MIME Media Type "application/ mbms-msk-response+xml" denotes that the message body contains the response codes to MSK request procedure in accordance with the XML scheme "urn:3GPP:metadata:2005:MBMS:mskResponse" (see sub-clause 11.8 and 3GPP TS 33.246 [20]).

Type name: application

Subtype name: mbms-msk-response+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The usage of the format is to identify one or more response codes of MSK requests procedures. Modification or changes to this format thus only result in a denial of service attack. Integrity protection would protect against such modifications. In MBMS this is accomplished using procedures described in 3GPP TS 33.246.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.13 Registration of MIME type "application/mbms-register-response+xml"

The MIME Media Type "application/mbms-register-response+xml" denotes that the message body contains the response codes to an MBMS User Service Registration or MBMS User Service De-Register procedure in accordance with the XML schema "urn:3GPP:metadata:2005:MBMS:securityRegistrationResponse" (see sub-clause 11.7 and 3GPP TS 33.246 [20]).

Type name: application

Subtype name: mbms-register-response+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The usage of the format is to identify one or more response codes of registration or deregistration procedures. Modification or changes to this format thus only result in a denial of service attack. Integrity protection would protect against such modifications. In MBMS this is accomplished using procedures described in 3GPP TS 33.246

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Magnus Westerlund (magnus.westerlund@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.14 Registration of MIME type "application/mbms-schedule+xml"

The MIME-Type registration for "application/mbms-schedule+xml" identifies XML documents that follows the schema " urn:3gpp:metadata:2011:MBMS:scheduleDescription " defined in clause 11.2A.2.

Type name: application

Subtype name: mbms-schedule+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

8bit.

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The information present in this media format is used to configure the receiving application. Thus the usage of the fomat is vulnerable to attacks modifying or spoofing the content of this format. It is recommended to use source authentication and integrity protection. The information carried in the media format does not contain any private information such as passwords or keys. The information carried in the media format does not contain executable content. Content from other namespaces may be introduced into this media type by extensions defined by 3GPP, and thus security considerations inherited from those extensions will need to be considered as new extensions are defined in future releases.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Eric Turcotte (Eric.Turcotte@ericsson.com)
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

C.15 Registration of MIME type "application/mbms-filter+xml"

The MIME-Type registration for "application/mbms-filter+xml" identifies XML documents that follows the schema " urn:3gpp:metadata:2011:MBMS:filterDescription " defined in clause 11.2B.

Type name: application

Subtype name: mbms-filter-description+xml

Required parameters:

None

Optional parameters:

charset: As specified in RFC 3023 [83] for media type application/xml.

Encoding considerations:

8bit

This is an XML document and the encoding considerations are the same as for media type "application/xml" defined in RFC 3023 [83].

Security considerations:

The information present in this media format is used to configure the receiving application. Thus the usage of the fomat is vulnerable to attacks modifying or spoofing the content of this format. It is recommended to use source authentication and integrity protection. The information carried in the media format does not contain any private information such as passwords or keys.

Interoperability considerations:

None

Published specification:

3GPP TS 26.346

Applications which use this media type:

3GPP MBMS based applications

Additional information:

Magic number(s): None

File extension(s):

Macintosh file type code(s):

Person & email address to contact for further information:

Charles Lo *(clo@qti.qualcomm.com)*
3GPP TSG SA WG4

Intended usage: COMMON

Restrictions on usage:

None

Author:

3GPP TSG SA WG4

Change controller:

3GPP TSG SA WG4

Annex D (informative): RTP packetization guidelines

This annex provides guidelines for MBMS senders to minimize initial buffering delay between starting of the reception and starting of rendering of media data in MBMS receivers.

When H.264 (AVC) video is in use, an MBMS sender should form FEC source blocks in which the first H.264 (AVC) access unit in decoding order is an IDR access unit.

MBMS senders should transmit all application data units for a given H.264 (AVC) access unit, or audio frame within one FEC source block.

MBMS senders should set the min-buffer-time MIME/SDP parameter and the minimum buffering delay elements included in FEC source blocks to values that are sufficient to cover any required de-interleaving of application data units, such as H.264 (AVC) NAL units and coded audio frames, from their transmission order to decoding order.

When RTP timestamps are converted to the wallclock time of the MBMS receiver, the smallest RTP timestamp among the FEC source packets of a FEC source block of a stream should be equal or close to the smallest RTP timestamp among the FEC source packets of a FEC source block of any other stream of the same MBMS streaming session.

When RTP timestamps are converted to the wallclock time of the MBMS receiver, the greatest RTP timestamp among the FEC source packets of a FEC source block of a stream should be equal or close to the greatest RTP timestamp among the FEC source packets of a FEC source block of any other stream of the same MBMS streaming session.

When DIMS content is used, suitable media level recovery (a random access point such as a replacement scene, redundant scene, etc.) should be present at the beginning of each FEC source block. This facilitates immediate rendering of the DIMS content after FEC decoding, thus reducing tune-in latency.

Annex E (informative): Void

Annex F (informative): Hybrid Streaming Delivery

F.1 Introduction

In hybrid streaming delivery scenarios it is assumed that a set of streaming services is available over MBMS and a set of streaming services is available over PSS. MBMS services may not be available in some service areas, in which cases those services might be alternatively provided via PSS.

Different switching cases between PSS and MBMS and vice versa may occur in these scenarios, including user-initiated content switching with access change as well as application-initiated access change. The latter can occur when a service is available over MBMS in some service areas but not available over MBMS in other service areas, assuming that the service is provided over PSS as a fall-back.

This annex describes some methods to improve switching times in such hybrid streaming delivery scenarios.

F.2 Switching between MBMS and PSS Access

F.2.1 Synchronization of Flows

For switching between MBMS and PSS access while receiving the same service, comparing the SSRC values of the PSS and MBMS flows gives the UE an advantage for synchronization onto the flows as described in sub-clause 8.5.

In order to allow for an as interruption-free access switch as possible, the time offset between MBMS packet reception and PSS packet reception (including MBMS FEC buffering and PSS retransmission delays) should be as small as possible.

It is advantageous (but not always possible) that the same set of codecs is used for representing the same service over both accesses. If the same media bit rates are provided over both accesses, then at best the same media flows (identical media encoding) are used for both accesses. This gives the UE an advantage of being able to continuously using decoding buffers thus simplifying seamless access switching. In case the codecs or codec level/profiles change upon access switching, the UE needs to prepare a new set of decoders and possibly having them run in parallel for a short period of time.

F.2.2 Switching from MBMS to PSS Access

Switching from MBMS to PSS can happen in two cases, user-initiated content switch with access change and application-initiated access change.

- User-initiated content switch with access change: The user requests reception of a service other than the one currently received over MBMS, and the new service is only available over PSS.
- Application-initiated switch of access: The MBMS coverage is lost and alternative reception of the same service is possible over PSS.

The latter case may occur suddenly and without warning when the available signal strength is no longer good enough for the MBMS reception. The UE may identify the loss of MBMS access by signal strength measurements, detection of packet losses exceeding a certain threshold, or that RTP reception has stopped completely. As a result the UE may terminate the MBMS session and initiate service reception over PSS instead.

In case MBMS coverage is lost and the corresponding PSS session is requested, the UE may use PSS time-shifting (as defined in TS 26.234 [47]) for signalling a PSS play-out start corresponding to the time instant the MBMS access was lost, giving the advantage of synchronization of the received flows.

F.2.3 Switching from PSS to MBMS Access

Switching from PSS to MBMS can happen in two cases, user-initiated content switch with access change and application-initiated access change.

- User-initiated content switch with access change: The user requests reception of a service other than the one currently received over PSS, and the new service is available over MBMS.
- Application-initiated switch of access: The UE recognizes that MBMS reception of that service is alternatively possible (e.g. because MBMS reception was re-gained), such that the UE may terminate the PSS session and initiate reception over MBMS.

If the UE can receive both PSS and MBMS flows at the same time, and the time offset between MBMS packet reception and PSS packet reception (including MBMS FEC buffering and PSS retransmission delays) is small, and identical media encoding is used, then application-initated access change from PSS to MBMS without service change is possible to be made in such a way that it is seamless.

Annex G (informative): Guidelines for Channel Tune-in and Switch Time Reduction

This Annex describes some methods to improve channel tune-in and switching times for MBMS Streaming when using the FEC framework. It presents means for MBMS senders and receivers to minimize delay from a channel switch and initial tune-in time. Fast startup improvements for the layers below IP are not described here.

G.1 Interleaving for Tune-in Time Reduction of FEC protected MBMS Services

Interleaving may be applied to source blocks before FEC encoding to re-arrange the order of transmission of the UDP packets. The target of the interleaving is to provide high priority and correctly decodeable media units in a way that maximizes the resulting media duration at the receivers that tune in at that specific source block. The interleaver may operate at two different levels:

- Inter-stream interleaving: prioritize the media streams and arrange their transmission order according to the ascending order of priorities. In other words high priority data, e.g. audio data, is transmitted towards the end of the source block.
- Intra-stream interleaving: high priority media data units such as Random Access Points of a video stream are transmitted towards the end of a source block.

The interleaving procedure enables receivers to reliably decode and present media data that has been received from a fraction of the tune-in source block.

Interleaving is transparent to legacy receivers.

G.1.1 Timestamp Offsets

The timestamp offset field may be used to signal a timestamp offset for the received media units in the FEC block as specified in 8.2.3.2.

The timestamp offsets may be used to reconstruct the presentation time line at the UE. They may also be used to reduce the out time caused by the reception of a partially received interleaved FEC source block as shown in figure G.1.

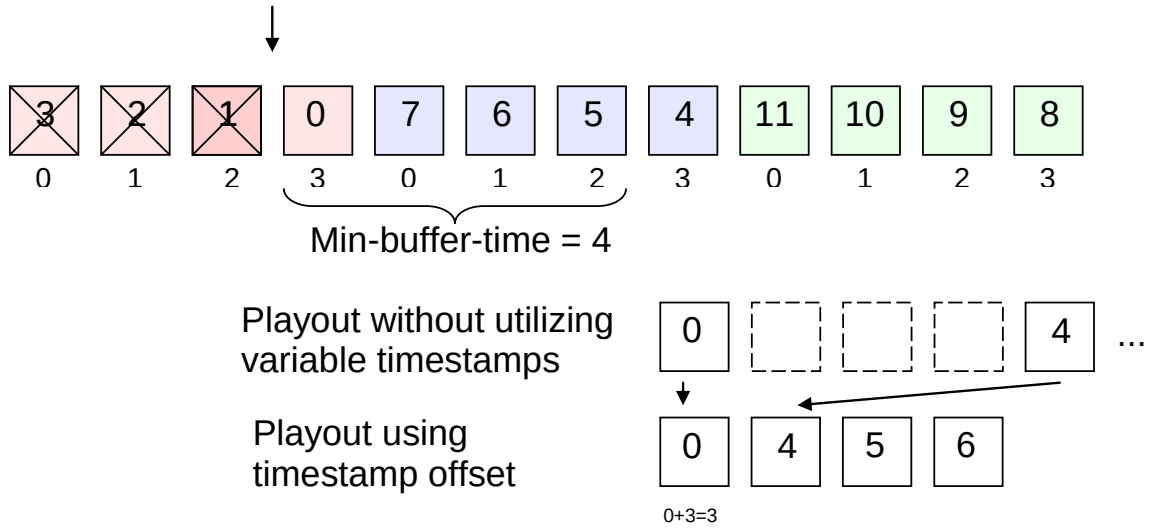


Figure G.1: Early tune-in using variable timestamps

G.1.2 Early Playout

A UE that desires to make use of the interleaving to reduce the tune-in time may start the playout earlier than dictated by the min-buffer-time value.

Instead, the UE may schedule the first media unit of the succeeding source block to be played out after the min-buffer-time. It may then estimate the appropriate time for starting early playout based on the amount of media duration that was received from the current block, the playout time of the earliest media unit of the next source block, and the highest presentation time of the media units of the current source block.

The early playout behavior is depicted by the following figure.

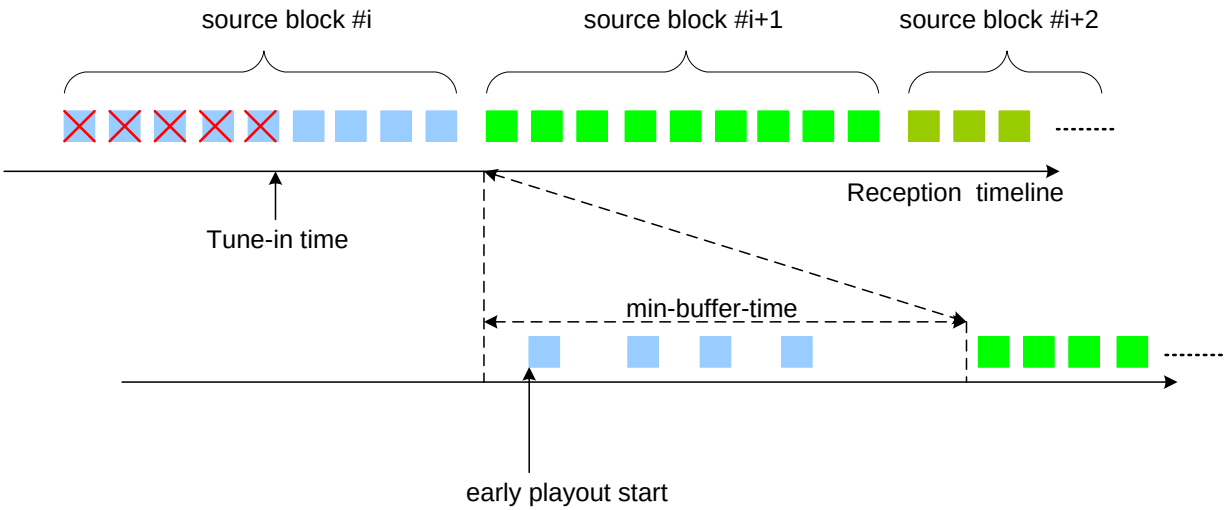


Figure G.2: Early play out of interleaved media data of a FEC protected MBMS service

G.2 FEC Stream & Channel Bundling

G.2.1 Introduction

FEC Stream bundling is a method of improving the FEC efficiency and also to improve channel switching times. Several flows of one or more user services are “bundled” to form the source blocks for the FEC calculations. . This means, that all flows must be received for potential FEC recovery.

If flows from more than one user service are bundled (i.e. Channel Bundling), then the receiver discards other services after FEC processing. When a switch is performed, media inside the same bundle is immediately available. Stream Bundling can be performed on whole channels, parts of channels, or not at all.

Note: FEC Bundling does not improve start-up times.

G.2.1.1 Full Channel Bundling (All flows of several user services)

When full bundling is used, more than one complete channel is bundled. A channel switch inside the bundle does not require rebuffering and can therefore be near instantaneous. However, full bundling requires the processing of all packets. In other words, at a single time instance all video streams and all audio streams are received and processed. Also, the number of channels possible is limited by the fixed bearer bandwidth.

G.2.1.2 No Bundling

When no bundling is used, each channel is protected separately. In a channel switch the new channel needs to be buffered for the full min-buffer-time. At a single time instant one video stream and one audio stream is received and processed.

G.2.1.3 Partial Channel Bundling (Some flows of several user services)

Partial bundling may be used to combine some of the benefits of full bundling and no bundling. In this case more than one parital channel is bundled. This may be used to reduce the processing required on the terminal while retaining fast channel switching between parts of the media.

Partial bundling may be used to bundle only the audio part of channels. In other words, at a single time instance one video stream and all audio streams are received. When a switch occurs the audio is switched instantaneously as it does not need to be rebuffered. The complexity of this case is substantially lower than full bundling and the number of channels which can be bundled given a fixed bearer bandwidth is substantially increased. This is due to the fact that the audio uses a substantially lower percentage of the bitrate compared to video.

G.2.1.3 Stream Bundling (All flows of a single user services)

When stream bundling is used, all flows of a single channel are bundled. The FEC source block is formed using audio, video and security data.

Annex H (informative): QoE Reporting Management Object Device Description Framework

This Device Description Framework (DDF) is the standardized minimal set. A vendor can define its own DDF for the complete device. This DDF can include more features than this minimal standardized version.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE MgmtTree PUBLIC "-//OMA//DTD-DM-DDF 1.2//EN"
"http://www.openmobilealliance.org/tech/DTD/dm_ddf-v1_2.dtd">
<MgmtTree>
   <VerDTD>1.2</VerDTD>
   <Man>--The device manufacturer--</Man>
   <Mod>--The device model--</Mod>
   <Node>
      <NodeName>3GPP_MBMSQOE</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
            <node/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <Scope>
            <Permanent/>
         </Scope>
         <DFTitle>The interior node holding all 3GPP MBMS QoE Metrics Reporting objects</DFTitle>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
      <Node>
         <NodeName>Enabled</NodeName>
         <DFProperties>
            <AccessType>
               <Get/>
            </AccessType>
            <DFFormat>
               <bool/>
            </DFFormat>
            <Occurrence>
               <One/>
            </Occurrence>
            <Scope>
               <Permanent/>
            </Scope>
            <DFTitle>The QoE reporting requested indicator</DFTitle>
            <DFType>
               <DDFName/>
            </DFType>
         </DFProperties>
      </Node>
      <Node>
         <NodeName>APN</NodeName>
         <DFProperties>
            <AccessType>
               <Get/>
            </AccessType>
            <DFFormat>
               <chr/>
            </DFFormat>
            <Occurrence>
               <ZeroOrOne/>
            </Occurrence>
            <DFTitle>The Access Point Name for QoE reporting</DFTitle>
            <DFType>
               <DDFName/>
            </DFType>
         </DFProperties>
```

```
</Node>
<Node>
   <NodeName>Format</NodeName>
   <DFProperties>
      <AccessType>
         <Get/>
      </AccessType>
      <DFFormat>
         <chr/>
      </DFFormat>
      <Occurrence>
         <ZeroOrOne/>
      </Occurrence>
      <DFTitle>The QoE metrics report format</DFTitle>
      <DFType>
         <DDFName/>
      </DFType>
   </DFProperties>
</Node>
<Node>
   <NodeName>Rules</NodeName>
   <DFProperties>
      <AccessType>
         <Get/>
      </AccessType>
      <DFFormat>
         <chr/>
      </DFFormat>
      <Occurrence>
         <One/>
      </Occurrence>
      <DFTitle>The QoE metrics rules</DFTitle>
      <DFType>
         <DDFName/>
      </DFType>
   </DFProperties>
</Node>
<Node>
   <NodeName>Session</NodeName>
   <DFProperties>
      <AccessType>
         <Get/>
      </AccessType>
      <DFFormat>
         <node/>
      </DFFormat>
      <Occurrence>
         <ZeroOrOne/>
      </Occurrence>
      <DFTitle>The QoE session metrics node</DFTitle>
      <DFType>
         <DDFName/>
      </DFType>
   </DFProperties>
   <Node>
      <NodeName>Metrics</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
            <chr/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
   </Node>
   <Node>
      <NodeName>Ext</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
```

```
            <node/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <Scope>
            <Permanent/>
         </Scope>
         <DFTitle> A collection of all extension objects</DFTitle>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
   </Node>
</Node>
<Node>
   <NodeName>Speech</NodeName>
   <DFProperties>
      <AccessType>
         <Get/>
      </AccessType>
      <DFFormat>
         <node/>
      </DFFormat>
      <Occurrence>
         <ZeroOrOne/>
      </Occurrence>
      <DFTitle>The QoE speech metrics node</DFTitle>
      <DFType>
         <DDFName/>
      </DFType>
   </DFProperties>
   <Node>
      <NodeName>Metrics</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
            <chr/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
   </Node>
   <Node>
      <NodeName>Ext</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
            <node/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <Scope>
            <Permanent/>
         </Scope>
         <DFTitle> A collection of all extension objects</DFTitle>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
   </Node>
</Node>
<Node>
   <NodeName>Video</NodeName>
   <DFProperties>
      <AccessType>
         <Get/>
      </AccessType>
      <DFFormat>
```

```
            <node/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <DFTitle>The QoE video metrics node</DFTitle>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
      <Node>
         <NodeName>Metric</NodeName>
         <DFProperties>
            <AccessType>
               <Get/>
            </AccessType>
            <DFFormat>
               <chr/>
            </DFFormat>
            <Occurrence>
               <ZeroOrMore/>
            </Occurrence>
            <DFType>
               <DDFName/>
            </DFType>
         </DFProperties>
      </Node>
      <Node>
         <NodeName>Ext</NodeName>
         <DFProperties>
            <AccessType>
               <Get/>
            </AccessType>
            <DFFormat>
               <node/>
            </DFFormat>
            <Occurrence>
               <ZeroOrOne/>
            </Occurrence>
            <Scope>
               <Permanent/>
            </Scope>
            <DFTitle>A collection of all extension objects</DFTitle>
            <DFType>
               <DDFName/>
            </DFType>
         </DFProperties>
      </Node>
   </Node>
   <Node>
      <NodeName>Text</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
            <node/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <DFTitle>The QoE timed text metrics node</DFTitle>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
      <Node>
         <NodeName>Metric</NodeName>
         <DFProperties>
            <AccessType>
               <Get/>
            </AccessType>
            <DFFormat>
               <chr/>
            </DFFormat>
            <Occurrence>
               <ZeroOrOne/>
            </Occurrence>
```

```
            <DFType>
               <DDFName/>
            </DFType>
         </DFProperties>
      </Node>
      <Node>
         <NodeName>Ext</NodeName>
         <DFProperties>
            <AccessType>
               <Get/>
            </AccessType>
            <DFFormat>
               <node/>
            </DFFormat>
            <Occurrence>
               <ZeroOrOne/>
            </Occurrence>
            <Scope>
               <Permanent/>
            </Scope>
            <DFTitle>A collection of all extension objects</DFTitle>
            <DFType>
               <DDFName/>
            </DFType>
         </DFProperties>
      </Node>
   </Node>
   <Node>
      <NodeName>Ext</NodeName>
      <DFProperties>
         <AccessType>
            <Get/>
         </AccessType>
         <DFFormat>
            <node/>
         </DFFormat>
         <Occurrence>
            <ZeroOrOne/>
         </Occurrence>
         <Scope>
            <Permanent/>
         </Scope>
         <DFTitle>A collection of all extension objects</DFTitle>
         <DFType>
            <DDFName/>
         </DFType>
      </DFProperties>
   </Node>
</Node>
</MgmtTree>
```

Annex I (informative): Void

Annex J (normative): Schema

J.1 User Service Description schema

This clause specifies the main USD Schema according to the current release. In addition to clause 11.2.1 specifying the USD schema content, additional requirements are included in this clause to ensure backward and forward compatibility.

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

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

NOTE 1: 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 a UE receives an instantiation of a USD compliant to this schema, it shall determine the USD schema version required to parse the USD instantiation as follows:

- If the UE supports one or more versions of the USD schema with the schema *version* attribute, then the UE shall use the USD schema that has the highest schema *version* attribute value that is equal to or less than the value in the received *schemaVersion* element;
- Otherwise, if the UE supports a USD schema without a schema *version* attribute, or all of its USD schemas with the *schema version* attribute have a value greater than the value received in the *schemaVersion* element, then the UE shall use its USD schema without a *version* attribute.

NOTE 2: To avoid a schema validation error (Unique Particle Attribution rule) if an optional element from another namespace is specified just prior to the *xs:any namespace="##other"* instructions (3 instances in the USD schema), a *schemaVersion* element is defined as the last child of the root element (*bundleDescription* element), and a *delimiter* element is specified preceding the 2 other occurrences of *xs:any* instructions. These 2 elements are defined in clause J.2, and the schema file name is “schema-version.xsd”.

The schema file names associated to the USD schemas specified in sub-clause 11.2.1 are as follows:

- Release 7 extension schema: USD-Rel-07-schema-snippet.xsd
- Release 8 extension schema: USD-Rel-08-schema-snippet.xsd
- Release 9 extension schema: USD-Rel-09-schema-snippet.xsd

The file name for the main USD schema (as defined below) is “USD-schema-main.xsd”.

User Service Description

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:r7="urn:3GPP:metadata:2007:MBMS:userServiceDescription"
    xmlns:r8="urn:3GPP:metadata:2008:MBMS:userServiceDescription"
```

```
    xmlns:r9="urn:3GPP:metadata:2009:MBMS:userServiceDescription"
    xmlns:sv="urn:3gpp:metadata:2009:MBMS:schemaVersion"
    targetNamespace="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
    elementFormDefault="qualified"
    version="1">

<xs:import schemaLocation="USD-Rel-07-schema-snippet.xsd"
namespace="urn:3GPP:metadata:2007:MBMS:userServiceDescription"/>
<xs:import schemaLocation="USD-Rel-08-schema-snippet.xsd"
namespace="urn:3GPP:metadata:2008:MBMS:userServiceDescription"/>
<xs:import schemaLocation="USD-Rel-09-schema-snippet.xsd"
namespace="urn:3GPP:metadata:2009:MBMS:userServiceDescription"/>
<xs:import schemaLocation="schema-version.xsd"
namespace="urn:3gpp:metadata:2009:MBMS:schemaVersion"/>

    <xs:element name="bundleDescription" type="bundleDescriptionType"/>
    <xs:complexType name="bundleDescriptionType">
        <xs:sequence>
            <xs:element name="userServiceDescription" type="userServiceDescriptionType"
maxOccurs="unbounded"/>
            <xs:element ref="r7:initiationRandomization" minOccurs="0"/>
            <xs:element ref="r7:terminationRandomization" minOccurs="0"/>
            <xs:element ref="sv:schemaVersion"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="fecDescriptionURI" type="xs:anyURI" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="userServiceDescriptionType">
        <xs:sequence>
            <xs:element name="name" type="nameType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="serviceLanguage" type="xs:language" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="requiredCapabilities" type="requirementsType" minOccurs="0"/>
            <xs:element name="deliveryMethod" type="deliveryMethodType" maxOccurs="unbounded"/>
            <xs:element name="accessGroup" type="accessGroupType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element ref="r7:serviceGroup" minOccurs="0"/>
            <xs:element ref="r7:initiationRandomization" minOccurs="0"/>
            <xs:element ref="r7:terminationRandomization" minOccurs="0"/>
            <xs:element ref="r8:Registration" minOccurs="0"/>
            <xs:element ref="r9:mediaPresentationDescription" minOccurs="0"/>
            <xs:element ref="r9:schedule" minOccurs="0"/>
            <xs:element ref="r9:availabilityInfo" minOccurs="0"/>
            <xs:element ref="sv:delimiter"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="serviceId" type="xs:anyURI" use="required"/>
        <xs:attribute ref="r7:serviceClass"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="accessGroupType">
        <xs:sequence>
            <xs:element name="accessBearer" type="xs:string" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="id" type="accessGroupIdType" use="required"/>
    </xs:complexType>
    <xs:complexType name="deliveryMethodType">
        <xs:sequence>
            <xs:element ref="r7:unicastAccessURI" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="r8:alternativeAccessDelivery" minOccurs="0"/>
            <xs:element ref="sv:delimiter"/>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="accessGroupId" type="accessGroupIdType" use="optional"/>
        <xs:attribute name="associatedProcedureDescriptionURI" type="xs:anyURI" use="optional"/>
        <xs:attribute name="protectionDescriptionURI" type="xs:anyURI" use="optional"/>
        <xs:attribute name="sessionDescriptionURI" type="xs:anyURI" use="required"/>
        <xs:attribute name="accessPointName" type="xs:anyURI" use="optional"/>
        <xs:anyAttribute processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="nameType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="lang" type="xs:language" use="optional"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
```

```
    <xs:simpleType name="accessGroupIdType">
        <xs:restriction base="xs:nonNegativeInteger"/>
    </xs:simpleType>
    <xs:complexType name="requirementsType">
        <xs:sequence>
            <xs:element name="feature" type="xs:unsignedInt" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

J.2 Version and Delimiter schema

This clause specifies the schema that is needed for the UE and the network side for forward and backward compatibility, and is used by the following schemas:

- USD
- Schedule Description
- Filter Description
- FDT

The schema defines 2 elements, *schemaVersion* and *delimiter*. The value of the *schemaVersion* is specified separately for each of the schemas above. The *delimiter* element is specified in the various main schemas above to avoid the schema validation error (due to the Unique Particle Attribution rule).

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

The file name used for this schema is “schema-version.xsd”, and is used in the various schemas above for the “import” instruction.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema  xmlns="urn:3gpp:metadata:2009:MBMS:schemaVersion"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            targetNamespace="urn:3gpp:metadata:2009:MBMS:schemaVersion"
            elementFormDefault="qualified">

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

</xs:schema>
```

Annex K (informative): Change history

Change history							
Date	**TSG SA#**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2005-03	27	SP-050082			Approved at TSG SA#27 Plenary	2.0.0	6.0.0
2005-06	28	SP-050250	001	1	Corrections to QoE metrics specification for MBMS	6.0.0	6.1.0
2005-06	28	SP-050250	002	1	Using two TMGIs	6.0.0	6.1.0
2005-06	28	SP-050250	003		MBMS Service Descriptions over HTTP	6.0.0	6.1.0
2005-06	28	SP-050250	004	1	Corrections to the specification of Associated Delivery Procedures for MBMS	6.0.0	6.1.0
2005-06	28	SP-050250	005	2	Usage of MBMS Session Identity	6.0.0	6.1.0
2005-06	28	SP-050250	010	1	MBMS user service announcement via point-to-point push bearers	6.0.0	6.1.0
2005-06	28	SP-050250	011		Removal of obsolete note	6.0.0	6.1.0
2005-06	28	SP-050250	013		Specification of Raptor Forward Error Correction and Streaming User Service bundling	6.0.0	6.1.0
2005-06	28	SP-050250	015		Clarification of Associated Delivery Procedure	6.0.0	6.1.0
2005-06	28	SP-050250	016		Corrections of FLUTE Support Requirements	6.0.0	6.1.0
2005-06	28	SP-050250	017		Corrections of the reference list	6.0.0	6.1.0
2005-06	28	SP-050250	018		Definition of RTP Session	6.0.0	6.1.0
2005-06	28	SP-050250	019		Corrections and editorial modifications to chapter 4	6.0.0	6.1.0
2005-06	28	SP-050250	020		MBMS Repair	6.0.0	6.1.0
2005-06	28	SP-050250	021		MBMS Media Codec Support	6.0.0	6.1.0
2005-09	29	SP-050423	0023	2	General corrections to MBMS	6.1.0	6.2.0
2005-09	29	SP-050423	0024	2	Corrections to MBMS download delivery procedure	6.1.0	6.2.0
2005-09	29	SP-050423	0025	3	FDT schema Correction	6.1.0	6.2.0
2005-09	29	SP-050423	0027	2	Specification of MIME types for MBMS	6.1.0	6.2.0
2005-09	29	SP-050423	0028	4	Correction of XML Schemas	6.1.0	6.2.0
2005-09	29	SP-050423	0029		Corrections to QoE metrics specification for MBMS	6.1.0	6.2.0
2005-09	29	SP-050423	0031	2	SDP Bug Fixes for the MBMS Download Delivery	6.1.0	6.2.0
2005-09	29	SP-050423	0035	1	Interpretation of TOI sequencing	6.1.0	6.2.0
2005-12	30	SP-050788	0034	1	MBMS User Service Announcement Application ID	6.2.0	6.3.0
2005-12	30	SP-050788	0037	3	Time Synchronization between BM-SCs and MBMS Ues	6.2.0	6.3.0
2005-12	30	SP-050788	0038	2	Reference and definition correction for MBMS user service	6.2.0	6.3.0
2005-12	30	SP-050788	0039	2	PtP file repair URI correction	6.2.0	6.3.0
2005-12	30	SP-050788	0040	2	Allocation of FEC Encoding IDs	6.2.0	6.3.0
2006-03	31	SP-060014	0041		Sender Current Time (SCT) and Expected Residual Time (ERT) Header Fields	6.3.0	6.4.0
2006-03	31	SP-060014	0042		FEC-OTI-FEC-Instance-ID support in MBMS	6.3.0	6.4.0
2006-03	31	SP-060014	0043	2	Update of AMR-WB+ RFC reference	6.3.0	6.4.0
2006-03	31	SP-060014	0045	1	Handling several Response Codes in one response message	6.3.0	6.4.0
2006-03	31	SP-060014	0046	1	Addition of a reference to TR 26.936	6.3.0	6.4.0
2006-06	32	SP-060352	0047	2	Clarification on FDT Instance data elements	6.4.0	6.5.0
2006-06	32	SP-060352	0048	1	Correction for the FEC block construction and example	6.4.0	6.5.0
2006-06	32	SP-060352	0049	1	MBMS Security function alignment	6.4.0	6.5.0
2006-06	32	SP-060352	0053	1	FEC correction	6.4.0	6.5.0
2006-06	32	SP-060357	0044	6	Scalable MBMS multicast session joining and leaving	6.5.0	7.0.0
2006-06	32	SP-060357	0050	1	Modification of MBMS User Service procedures to enable unicast bearer usage	6.5.0	7.0.0
2006-06	32	SP-060357	0051	1	Modification of MBMS User Service architecture to enable unicast bearer usage	6.5.0	7.0.0
2006-09	33	SP-060593	0057	2	Editorial modification (deletion of duplicated text)	7.0.0	7.1.0
2006-09	33	SP-060593	0059	1	Editorial Improvements and a correction of the MBMS FEC	7.0.0	7.1.0
2006-12	34	SP-060848	0055	2	Correction to the file repair request	7.1.0	7.2.0
2006-12	34	SP-060848	0061	1	Essential Correction of FLUTE FDT Content Encoding Support	7.1.0	7.2.0
2006-12	34	SP-060848	0063	1	Essential Correction of SDP for Streaming Session	7.1.0	7.2.0
2006-12	34	SP-060848	0065	2	Correction of required number of FEC Repair Symbols	7.1.0	7.2.0
2006-12	34	SP-060848	0067	1	Correction of declaration for ‘xs:anyAttribute’ and of missing opening tag ‘<keyDomainID>’	7.1.0	7.2.0
2006-12	34	SP-060848	0069	3	Inclusion of the MBMS Counting Indication to the MBMS bearer mode description	7.1.0	7.2.0
2007-03	35	SP-070024	0072	2	Essential correction for the support of multiple versions in file repair	7.2.0	7.3.0
2007-03	35	SP-070027	0073	3	Caching Directive in MBMS	7.2.0	7.3.0
2007-03	35	SP-070024	0075	1	Essential correction of a wrong Reference	7.2.0	7.3.0
2007-03	35	SP-070027	0077	2	MBMS Download in Roaming Condition	7.2.0	7.3.0
2007-03	35	SP-070024	0080	1	Essential Corrections of the Reception Reporting Procedure	7.2.0	7.3.0

Change history							
Date	**TSG SA#**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2007-03	35	SP-070027	0081	4	Hybrid PSS / MBMS Streaming services and Corrections to the MBMS User Service Description Schema	7.2.0	7.3.0
2007-03	35	SP-070027	0082		FLUTE session set-up with RTSP	7.2.0	7.3.0
2007-06	36	SP-070317	0084	1	Correction of termination of unicast bearer service based services	7.3.0	7.4.0
2007-06	36	SP-070315	0086	1	Correction of references in MBMS	7.3.0	7.4.0
2007-06	36	SP-070319	0089	3	Inclusion of DIMS in MBMS	7.3.0	7.4.0
2007-06	36	SP-070317	0090	2	Signalling of Initial Buffering Period	7.3.0	7.4.0
2007-06	36	SP-070317	0091	2	MBMS download service delivery to UEs in roaming condition	7.3.0	7.4.0
2007-06	36	SP-070317	0092	2	Scalability extensions for unicast delivery of MBMS services	7.3.0	7.4.0
2007-06	36	SP-070317	0093		Optimization to allow a smooth transition between MBMS and PSS	7.3.0	7.4.0
2007-06	36	SP-070315	0095	2	Correction of HTTP Response Error Codes	7.3.0	7.4.0
2007-06	36	SP-070317	0096	1	Addition of HTTP Response Error Codes	7.3.0	7.4.0
2007-06	36	SP-070317	0097		Improved video support for MBMS	7.3.0	7.4.0
2007-09	37	SP-070630	0078	3	Service Class Handling	7.4.0	7.5.0
2007-09	37	SP-070629	0088	3	Clarification of APN association and addition of APN elements	7.4.0	7.5.0
2007-09	37	SP-070629	0100		Correcting TMGI signalling and interpretation in SDP	7.4.0	7.5.0
2007-09	37	SP-070629	0102	1	Correction of several ABNF definitions	7.4.0	7.5.0
2007-09	37	SP-070630	0103		Correction of the OMA Push application id	7.4.0	7.5.0
2007-09	37	SP-070630	0104		Correction of QoE handling for MBMS Streaming on unicast	7.4.0	7.5.0
2007-09	37	SP-070630	0107	1	Clarification to the MBMS to PSS handover	7.4.0	7.5.0
2007-09	37	SP-070630	0108	1	Reception reporting by roaming MBMS users	7.4.0	7.5.0
2007-09	37	SP-070630	0109	2	Further correction to Required Service Capability for Service Access	7.4.0	7.5.0
2007-09	37	SP-070629	0111	1	Correction to ABNF syntax of QoE metrics in MBMS	7.4.0	7.5.0
2007-09	37	SP-070629	0113	1	Clarification of HTTP Response Error Codes for File Repair	7.4.0	7.5.0
2007-09	37	SP-070630	0114	1	Clarification of Release 7 specific HTTP Response Error Codes for File Repair	7.4.0	7.5.0
2007-12	38	SP-070762	0116	2	MBSFN bearer mode signalling	7.5.0	7.6.0
2007-12	38	SP-070762	0117	1	Clarification of re-synchronization after switch to unicast delivery	7.5.0	7.6.0
2008-03	38	SP-080010	0119	1	Corrections to different MBMS XML Schemas	7.6.0	7.7.0
2008-03	38	SP-080010	0121		Clarification of the reception reporting procedure start time	7.6.0	7.7.0
2008-06	40	SP-080251	0123	1	Correction to caching directive signaling	7.7.0	7.8.0
2008-09	41	SP-080468	0125	1	H.264 MIME/SDP reference correction	7.8.0	7.9.0
2008-09	41	SP-080468	0127	2	Correction to QoE reception reporting	7.8.0	7.9.0
2008-09	41	SP-080468	0130	2	Correcting file repair response for file-group requests	7.8.0	7.9.0
2008-09	41	SP-080477	0128	3	Enhancements of the QoE feature in MBMS	7.9.0	8.0.0
2008-09	41	SP-080477	0129	2	Registration procedure for MBMS User Service consumption	7.9.0	8.0.0
2008-09	41	SP-080477	0132	2	Transmitting a Full FDT snapshot	7.9.0	8.0.0
2008-12	42	SP-080681	0135	2	Time-shifting capability indication for MBMS services	8.0.0	8.1.0
2008-12	42	SP-080681	0137	1	Updates of references in MBMS	8.0.0	8.1.0
2008-12	42	SP-080681	0138	2	Hybrid streaming delivery appendix	8.0.0	8.1.0
2008-12	42	SP-080681	0141	1	Corrections of the FEC framework for MBMS Streaming	8.0.0	8.1.0
2008-12	42	SP-080681	0142		Additional guideline to use FEC streambundling for MBMS	8.0.0	8.1.0
2008-12	42	SP-080681	0143	1	Interleaving for fast channel switching and tune-in time in FEC-protected MBMS services	8.0.0	8.1.0
2009-03	43	SP-090004	0146		Addition of Timed Text payload format	8.1.0	8.2.0
2009-03	43	SP-090004	0149		Error in SDP for QoE configuration	8.1.0	8.2.0
2009-06	44	SP-090259	0153		Corrections for MBMS HSPA evolution	8.2.0	8.3.0
2009-06	44	SP-090256	0154		Correction of the MBMS User Service registration procedure	8.2.0	8.3.0
2009-09	45	SP-090567	0159	1	Clean-up corrections	8.3.0	8.4.0
2009-09	45	SP-090572	0155	2	QoE Alignment for MBMS	8.4.0	9.0.0
2009-09	45	SP-090572	0156	1	Key indicating of protected download data in FLUTE FDT	8.4.0	9.0.0
2009-09	45	SP-090572	0158	1	Using separate MBMS bearers for UTRAN and E-UTRAN in EPS	8.4.0	9.0.0
2009-12	46	SP-090716	0160		Correction of unreadable picture	9.0.0	9.1.0
2009-12	46	SP-090710	0161	1	MBMS DDF for QoE	9.0.0	9.1.0
2009-12	46	SP-090710	0162	1	MBMS QoE reporting during buffering periods	9.0.0	9.1.0
2009-12	46	SP-090711	0163	2	Video profile and level updates	9.0.0	9.1.0
2010-03	47	SP-100024	0165	1	HTTP Streaming with MBMS	9.1.0	9.2.0
2010-06	48	SP-100298	0167	1	Correcting the Access Bearer Identifier for different MBSFNs (IMB, TDD and FDD)	9.2.0	9.3.0
2010-06	48	SP-100303	0168	1	Addition of Timed Graphics to MBMS	9.2.0	9.3.0
2010-09	49	SP-100466	0169	2	Clarification on User Service Announcement over a MBMS bearer	9.3.0	9.4.0
2011-03	51				Inclusion of LTE logo	9.4.0	9.4.1
2011-03	51				Version for Release 10	9.4.1	10.0.0
2011-09	53	SP-110550	0171	2	Time Synchronisation between UE and BM-SC	10.0.0	10.1.0
2011-09	53	SP-110550	0172	2	Clarification of the TMGI Format	10.0.0	10.1.0
2011-09	53	SP-110550	0173	2	Missing QoE Attributes For FLUTE Transport	10.0.0	10.1.0
2011-09	53	SP-110550	0174	1	On counting clarifications	10.0.0	10.1.0
2011-11	54	SP-110795	0170	5	Initialisation Segment incorporation in DASH over MBMS	10.1.0	10.2.0
2011-11	54	SP-110789	0180	1	AlternativeUnicastDelivery	10.1.0	10.2.0

Change history							
Date	**TSG SA#**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2011-11	54	SP-110792	0186	2	Clarification of QoE applicability	10.1.0	10.2.0
2011-11	54	SP-110792	0191	2	Correction to MBMS Session Identity and EPS support	10.1.0	10.2.0
2011-11	54	SP-110792	0193	2	File Repair Error Code	10.1.0	10.2.0
2012-03	55	SP-120023	0197	1	DASH Layer QoE	10.2.0	10.3.0
2012-03	55	SP-120020	0200	1	Addition of complete FDT & USD schema	10.2.0	10.3.0
2012-03	55	SP-120020	0202		Correction of StaR-only sample percentage, backoff timer initialisation and RAck timer initialisation in MBMS reception reporting mechanism	10.2.0	10.3.0
2012-03	55	SP-120023	0203	1	TMGI & mbms-counting-indication correction	10.2.0	10.3.0
2012-03	55	SP-120023	0207	3	Clarification on 3GPP File Format Support for MBMS	10.2.0	10.3.0
2012-03	55	SP-120020	0209	1	Miscellaneous corrections	10.2.0	10.3.0
2012-03	55	SP-120020	0211	1	MBMS Scheduling Information	10.2.0	10.3.0
2012-03	55	SP-120020	0213		Aligning security parameters in Service Protection Description with SA3 MBMS TS	10.2.0	10.3.0
2012-03	55	SP-120026	0194	3	On MBMS video enhancements	10.3.0	11.0.0
2012-03	55	SP-120026	0195	2	USD Signaling of Frequency and Service Area Information for Service Continuity	10.3.0	11.0.0
2012-03	55	SP-120026	0215	4	Reception Reporting from Specific UEs	10.3.0	11.0.0
2012-06	56	SP-120219	0227	1	Reception reporting corrections	11.0.0	11.1.0
2012-06	56	SP-120219	0230		MBMS USD Schema correction	11.0.0	11.1.0
2012-06	56	SP-120219	0234	1	MBMS Schedule Info corrections	11.0.0	11.1.0
2012-06	56	SP-120224	0238	2	Reception Report Aggregation	11.0.0	11.1.0
2012-06	56	SP-120224	0240	2	MBMS File Repair Using Conventional HTTP Servers	11.0.0	11.1.0
2012-06	56	SP-120219	0257	4	MBMS Schema Compatibility Correction	11.0.0	11.1.0
2012-06	56	SP-120219	0260	1	MBMS Metadata Envelope Correction	11.0.0	11.1.0
2012-09	57	SP-120498	0248	4	Updates to TS 26.346 referring to IETF FECFRAME	11.1.0	11.2.0
2012-09	57	SP-120504	0261	2	Server Selection for MBMS File Repair via Conventional HTTP Servers	11.1.0	11.2.0
2012-09	57	SP-120504	0262	1	MD5 Procedures for MBMS File Repair via Conventional HTTP Servers	11.1.0	11.2.0
2012-09	57	SP-120504	0267	1	EMM-DDE file repair overload retry timer	11.1.0	11.2.0
2012-09	57	SP-120498	0273	2	MBMS Reception Reporting and USD Schema Correction	11.1.0	11.2.0
2012-09	57	SP-120498	0276		MBMS FDT Instance ID Schema correction for OMA Push Bearer	11.1.0	11.2.0
2012-09	57	SP-120504	0280		Reception Report Aggregation	11.1.0	11.2.0
2012-09	57	SP-120504	0281	1	Pre-FEC QoE Metric for Download Delivery Sessions	11.1.0	11.2.0
2012-09	57	SP-120509	0282	2	Frame Compatible Stereoscopic 3D video support	11.1.0	11.2.0
2012-09	57	SP-120508	0285	1	USD Updates on 'availabilityInfo' Description	11.1.0	11.2.0
2012-12	58	SP-120761	0265	3	Support for Location Filtering	11.2.0	11.3.0
2012-12	58	SP-120761	0266	6	Partial File Delivery via FLUTE	11.2.0	11.3.0
2012-12	58	SP-120757	0293	2	Adding LTE network resource	11.2.0	11.3.0
2012-12	58	SP-120761	0298		MBMS Reception Reporting Schema Correction	11.2.0	11.3.0
2012-12	58	SP-120757	0301	1	MBMS Schedule fragment Examples	11.2.0	11.3.0
2012-12	58	SP-120764	0304	1	MBMS UE UTC Time Synchronization	11.2.0	11.3.0
2012-12	58	SP-120761	0305	2	Correction to Byte Range-based File Repair	11.2.0	11.3.0
2013-03	59	SP-130016	0264	6	Encrypted DASH contents delivered over MBMS	11.3.0	11.4.0
2013-03	59	SP-130020	0307	1	MBMS Filter Description fragment MIME Type	11.3.0	11.4.0
2013-03	59	SP-130014	0310		MBMS Schedule fragment MIME Type	11.3.0	11.4.0
2013-03	59	SP-130014	0313		MBMS Schedule Examples Correction	11.3.0	11.4.0
2013-03	59	SP-130014	0316	2	Correction to MBMS bearer identification	11.3.0	11.4.0
2013-03	59	SP-130014	0319	2	Extend unicast bearer service to E-UTRAN	11.3.0	11.4.0
2013-06	60	SP-130187	0306	2	Confidentiality Protection of Associated Delivery Procedure Messages	11.4.0	11.5.0
2013-06	60	SP-130183	0323	2	Correction of MBMS Session Identity	11.4.0	11.5.0
2013-06	60	SP-130183	0326	1	Correction to metadata reference and USD example	11.4.0	11.5.0
2013-09	61	SP-130349	0327		Correction to Reception Reporting Aggregation postReceptionReport timer calculation	11.5.0	11.6.0
2013-09	61	SP-130346	0330		Essential Correction to Schedule Fragment MIME Type	11.5.0	11.6.0
2013-09	61	SP-130348	0332	1	Essential Corrections of DASH over MBMS	11.5.0	11.6.0
2013-12	62	SP-130562	0339	3	USD schema correction for security protection and URN	11.6.0	11.7.0
2013-12	62	SP-130562	0342	2	USD schema correction for OMA push usage for MBMS download	11.6.0	11.7.0
2013-12	62	SP-130566	0344	1	ClientID in RAck Reception Reporting	11.6.0	11.7.0
2013-12	62	SP-130568	0349		USD Schema Bug Fix	11.6.0	11.7.0
2013-12	62	SP-130566	0351		Correction to Partial File Delivery from MBMS Receiver to DASH Client	11.6.0	11.7.0