ETSI TR 126 946 V9.0.1 (2011-05)

Technical Report

Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Multimedia Broadcast/Multicast Service (MBMS) user service guidelines (3GPP TR 26.946 version 9.0.1 Release 9)

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Reference

RTR/TSGS-0426946v901

Keywords

GSM, UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Foreword

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Contents

Foreword

This Technical Report 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 user services can be built on top of the MBMS bearer service. The present document describes the usage of the two delivery methods, which are defined in [6]. The two delivery methods are streaming and download. 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.

The objective of the present document is to provide an overview of the MBMS System, and describes how the MBMS User Services use the MBMS Bearer Services.

1 Scope

The present document is only informative. In case there are any contradiction between the present document and 26.346, then the 3GPP TS 26.346 takes precedence.

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 (MBMS); 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 the Multimedia Broadcast/Multicast Service (MBMS) in the Radio Access Network (RAN); Stage 2".

[6] 3GPP TS 26.346: "Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs".

[7] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic bootstrapping architecture".

[8] 3GPP TS 33.246: "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)".

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

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

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

[12] IETF RFC 3452 (December 2002): "Forward Error Correction (FEC) Building Block", M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley and J. Crowcroft.

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

[14] IETF RFC 2327 (April 1998): "SDP: Session Description Protocol", M. Handley and V. Jacobson.

[15] IETF RFC 2357 (June 1998): "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols", A. Mankin, A. Romanow, S. Bradner and V. Paxson.

[16] IETF RFC 3550 (July 2003): "RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne, S. Casner, P. Frederick and V. Jacobson.

[17] ITU-T Recommendation H.264 (2003): "Advanced video coding for generic audiovisual services" | ISO/IEC 14496-10:2003: "Information technology - Coding of audio-visual objects - Part 10: Advanced Video Coding".

[18] IETF RFC 3984 (February 2005): "RTP payload Format for H.264 Video", S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund and D. Singer.

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 service: See 3GPP TS 22.146 [2].

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

Local MBMS Area: area of a MBMS service, where the service content is the same. One MBMS service may have different content in different local broadcast areas

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

MBMS user services: 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
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: MBMS user service initiation refers to the UE mechanism to set up the reception the MBMS user service data

MBMS delivery method: mechanism used by a MBMS user service to deliver content
There are two MBMS delivery method instances: download and streaming.

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

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

multicast service: See 3GPP TS 22.146 [2].

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

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
ERT	Expected Residual Time
ESI	Encoding Symbol ID
FDT	File Delivery Table
FEC	Forward Error Correction
FLUTE	File deLivery over Unidirectional Transport
GGSN	Gateway GPRS Serving Node
GPRS	General Packet Radio Service
IP	Internet Protocol
LCT	Layered Coding Transport
MBMS	Multimedia Broadcast/Multicast Service
MIME	Multipurpose Internet Mail Extensions
MS	Mobile Station
MSK	MBMS Service Key
MTK	MBMS Traffic Key
MUK	MBMS User Key
PSS	Packet Switch Streaming
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
TMGI	Temporary Mobile Group Identity
TOI	Transport Object Identifier
TSI	Transport Session Identifier
UDP	User Datagram Protocol
UE	User Equipment
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	eXtensible Markup Language

4 Overview

4.1 Phasing model

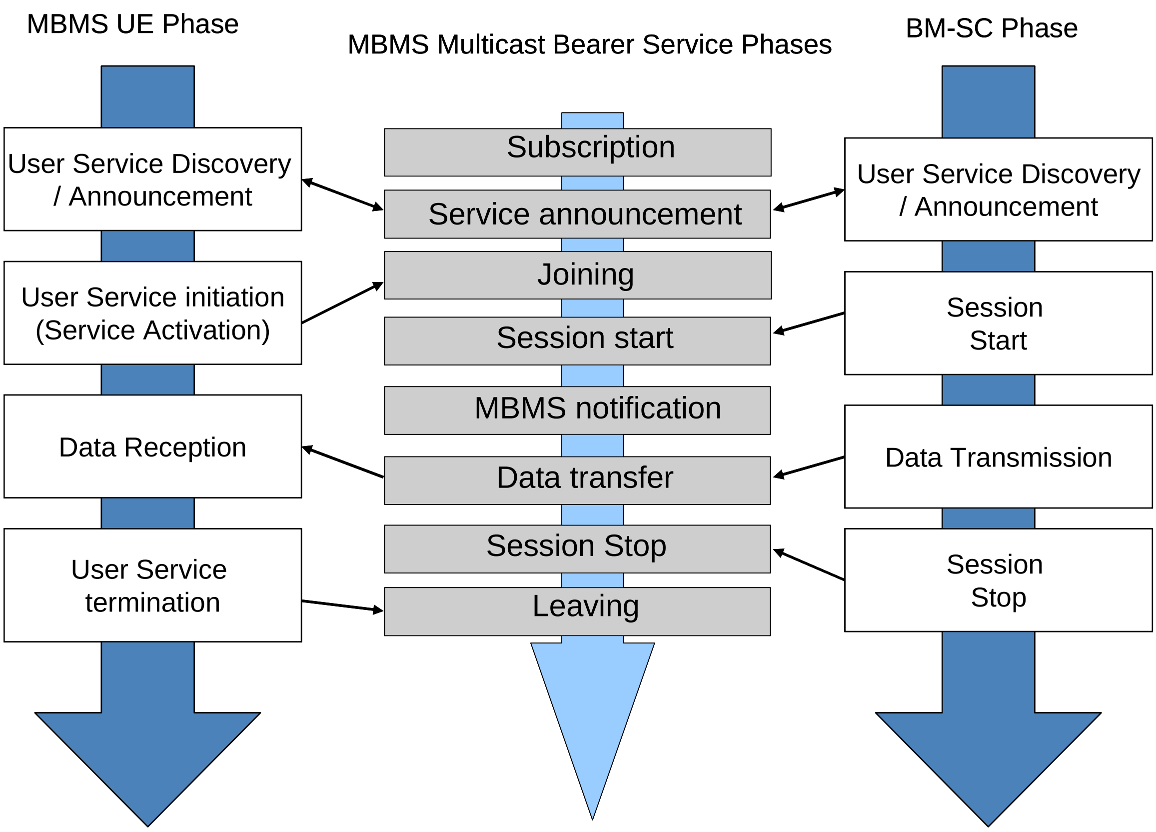


Figure 1: Mapping between MBMS Bearer and User Service phases

This mapping is applicable for the MBMS Broadcast and Multicast Modes.

It is assumed that the MBMS bearer service phases "Session start" and "MBMS notification" are handled autonomously and without interaction from the receiver application perspective.

It is assumed, that the Session Start phase "triggers" MBMS Notification phase.

4.2 System overview

Figure 2 gives an overview of functions for MBMS user services based on the MBMS download and the MBMS streaming delivery methods. The figure includes the MBMS User service provision phases (arrow on the right side of the figure). The Phase "MBMS User Service activation / deactivation" is an individual phase and is generally triggered by a user action. The break in the sequence indicates that the service activation phases are independent from the session start/stop and the data transfer phases. The service provider initiates the establishment of the MBMS User Plane (Session Start/Stop and the Data Transfer Phases) when there is content to be transmitted.

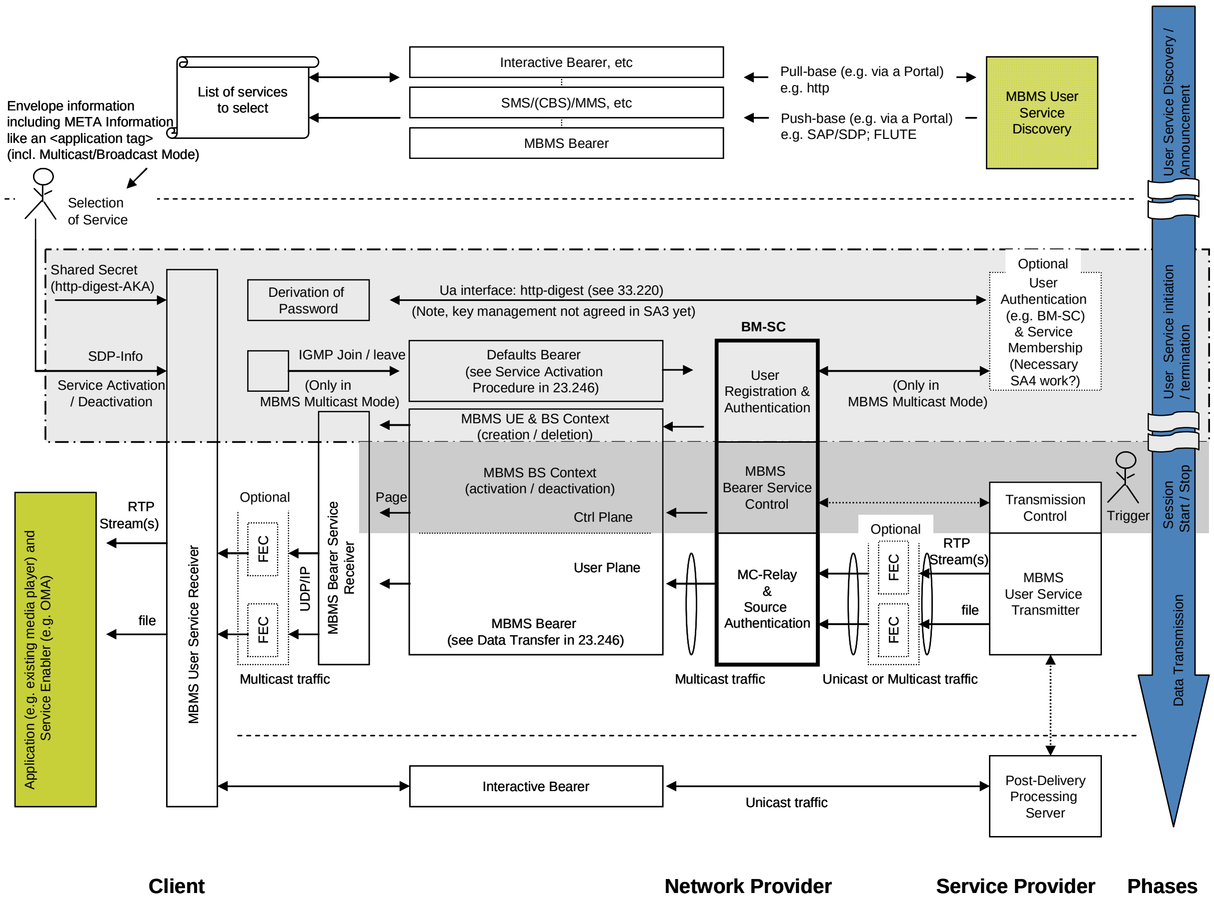


Figure 2: MBMS system overview

The MBMS Bearer Services are embedded in MBMS User Service procedures. The major part of the data is transferred via the MBMS Bearer Service. MBMS User Services include Associated-Delivery procedures (e.g. file repair) to augment the original MBMS Bearer sessions. Security functions are incorporated into the overview figure. One dedicated user authentication procedure is depicted in conjunction with the MBMS Bearer Service Activation procedure within the Session Establishment phase.

The figure contains client and network functions. Interactions, which are only applicable for the MBMS Multicast Mode, are marked accordingly.

5 Description of User Service procedures

5.1 User Service Discovery / Announcement

The "User Service Discovery / Announcement" phase provides all necessary information about available services and needed parameters to become member of a service. User Service Discovery / Announcement mechanism can provide information on available MBMS User services in pull mode, via the Web or WAP Portals, or in push mode, via SMS or MBMS Download based User Service.

The User Service Discovery / Announcement phase provides all necessary information for the "Service Activation" triggered by the UE. The necessary session parameters to configure the service activation are provided according to Session Description Protocol (SDP) specification (RFC 2327 [14]), which are complemented by additional information containing additional service parameters. Additionally, Service Discovery includes DRM/security descriptions.

In the case of a size-limited delivery method (e.g. SMS, MMS) being used during the Service Announcement / Discovery phase, it should be recognized that Service Announcement messages are not of fixed length. Methods of limiting the size of the message may be adopted. It is suggested that should a size-limited delivery method be selected, parameters such as URI lengths and the number of Connection data and Media announcement fields should be chosen appropriately.

5.2 User Service Initiation / termination procedure

For MBMS User service activation, the UE needs to perform a security function and the MBMS Bearer Service activation procedures. In the case of MBMS Multicast Mode, an IGMP or MLD message is sent via a "default" PDP context and triggers the creation of the MBMS UE context in various nodes. A MBMS Bearer Service (BS) context is only created once when the very first MBMS UE context is created in a node. In the case of MBMS Broadcast Mode, network-side elements establish the MBMS BS context without requiring user initiated messaging. The basic idea behind these procedures is that several UEs share one single bearer. The IP Multicast address and information for the MBMS User Service receiving application is given via the MBMS Service Discovery mechanism.

In case the service requires user authentication, security procedures are performed before the MBMS bearer Service Activation procedure. The "user authentication" procedure is a service optional security procedure. A password is derived from the shared secret (result of a GBA run) and used in the http-digest security procedure to authenticate the user (see 3GPP TS 33.220 [7] and 3GPP TS 33.346 [8] for further details).

The "User Authentication & Service Membership" phase provides procedures to register and authenticate users. The procedures shall follow the security specifications. In the case of MBMS Multicast Mode, this phase may authorize the service activation request from the UE. This function is present when security procedures like authentication and traffic encryption are required.

5.3 Session Start / Stop

According to the 3GPP TS 23.246 [4], the BM-SC controls the activation and the release of the MBMS User plane (switching of the MBMS BS context between "Active" and "Idle"). The Service Provider might trigger this process on the availability of new content. The release of the user plane resources depends on the transmission duration of the content. The transmission duration depends in case of the MBMS Download User Service on the content size, on the (optional) Forward-Error-Correction (FEC) overhead for error protection and the bitrate of the MBMS bearer.

5.4 Data transmission

The "MBMS User Service Transmitter" contains the MBMS User Service specific transmission protocols. Optionally, the content is protected by a Forward Error Correction (FEC) code. The traffic is sent using either IP unicast addressing or IP Multicast addressing via the BM-SC onto the MBMS Bearer Service. Note, the BM-SC is able to take IP Unicast and IP Multicast traffic as input according to 3GPP TS 23.246 [4].

In case of IP Unicast, the BM-SC is directly addressed by the MBMS User Service Transmitter and relays the traffic to IP Multicast. In case of IP Multicast, the traffic is forwarded by the BM-SC and the MBMS User Service Transmitter forwards the data to an IP Multicast group. Additionally the MC-Relay in the BM-SC may perform additional source authentication procedures to avoid unauthorized traffic entering the Core Network.

The "MBMS User Service Receiver" combines the reception via the MBMS Bearer Service and interactive bearer services in a controlled way. Optional "Associated-Delivery Processing Servers" are invoked after the actual MBMS data reception from the clients. A typical associated-delivery procedure is a point-to-point file repair mechanism for delivering missing data to clients of an MBMS Download session, which perceived packet losses that are unrecoverable during the broadcast/multicast transmission. The load of the point-to-point repair mechanism may be spread randomly in time and also across network elements. The MBMS Bearer Service may be configured to provide a constant or variable bit rate as requested by the MBMS User Service. The MBMS Bearer Service is expected to be service independent and can be configured by a Network Provider for MBMS Download and MBMS Streaming User Services.

5.5 Associated-Delivery procedures

Associated-delivery procedures may also include the usage of MBMS bearers.

5.5.1 File Repair service

It should be possible for UEs to repair erroneous files, delivery by the download delivery method, by means of repair request and response operations.

5.5.2 Content reception reporting

It should be however possible for the operator to collect statistical data such as lost frames, assigned resources, bit-rates achieved, etc. (3GPP TS 22.246 [3]).

6 Delivery methods

6.1 Download delivery method

The MBMS Download Delivery Method allows the error-free transmission of files via the unidirectional MBMS Bearer Services. The files are "downloaded" and stored in the local files-system of the user equipment. The network triggers the transmission since the users are registered to the download service. There is no further user request necessary after the service registration. Files may contain multimedia components or any other binary data. The MBMS Download Delivery Method allows the transmission of an arbitrary number of files within a single data transfer phase.

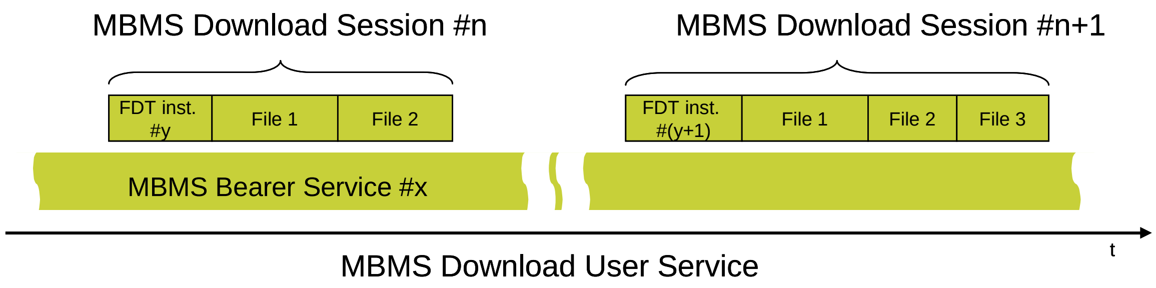


Figure 3: Definition of MBMS Download Sessions

Figure 4 is an example of an MBMS User Service based on the Download Delivery Method. The file transmission events are organized in MBMS Download Sessions. Each session is started with a File Delivery Table (FDT) instance, which describes in this example each file within the MBMS Download Session in terms of file name and file type (MIME Content Type). The service operator and the actual service would determine the timing of MBMS Download Sessions.

Users of the download-based MBMS User Service are not required to recognize the MBMS Downloading process itself, the reception may be happening completely in background. The user may be only informed about the completion of a MBMS download and that there is new content available on the terminal. Depending on the service, the MBMS Download session may require time-critical delivery of content.

6.1.1 SDL diagram of the download delivery method for the MBMS UE

Figure 4 and figure 5 describes the operation of an MBMS download receiver. The MBMS UE performs the steps described in the SDL diagram for each received Transport Object. An MBMS FDT may include a number of transmission objects, each identified by a unique Transmission Object Identifier (TOI).

The SDL identifies four states of the MBMS UE. A brief description of these states follows:

- The "**standby**" state reflects the state of the MBMS UE, in which MBMS download receiver is initialized and bound to one specific download channel (IP Multicast Address/Port plus Source Address). Note, the MBMS bearer is not in active state.
- The "**Object Reception**" state reflects the state of the MBMS UE, in which the MBMS UE is receiving data for transport objects. The Transport Object Identifier (TOI) identifies one transport object. The UE may receive FDT Instance updates related to the object being downloaded, in which case it should update the corresponding information for that object.
- The "**defer file repair**" state reflects the state of the MBMS UE, in which the MBMS UE deferring the file repair request message. Deferring the file repair request message is described in clause 9.3.4 of 3GPP TS26.346 [6].
- The "**defer reception reporting**" state reflects the state of the MBMS UE, in which the MBMS UE deferring the file repair request message. Deferring the file repair request message is described in clause 9.4.4 of 3GPP TS 26.346 [6].

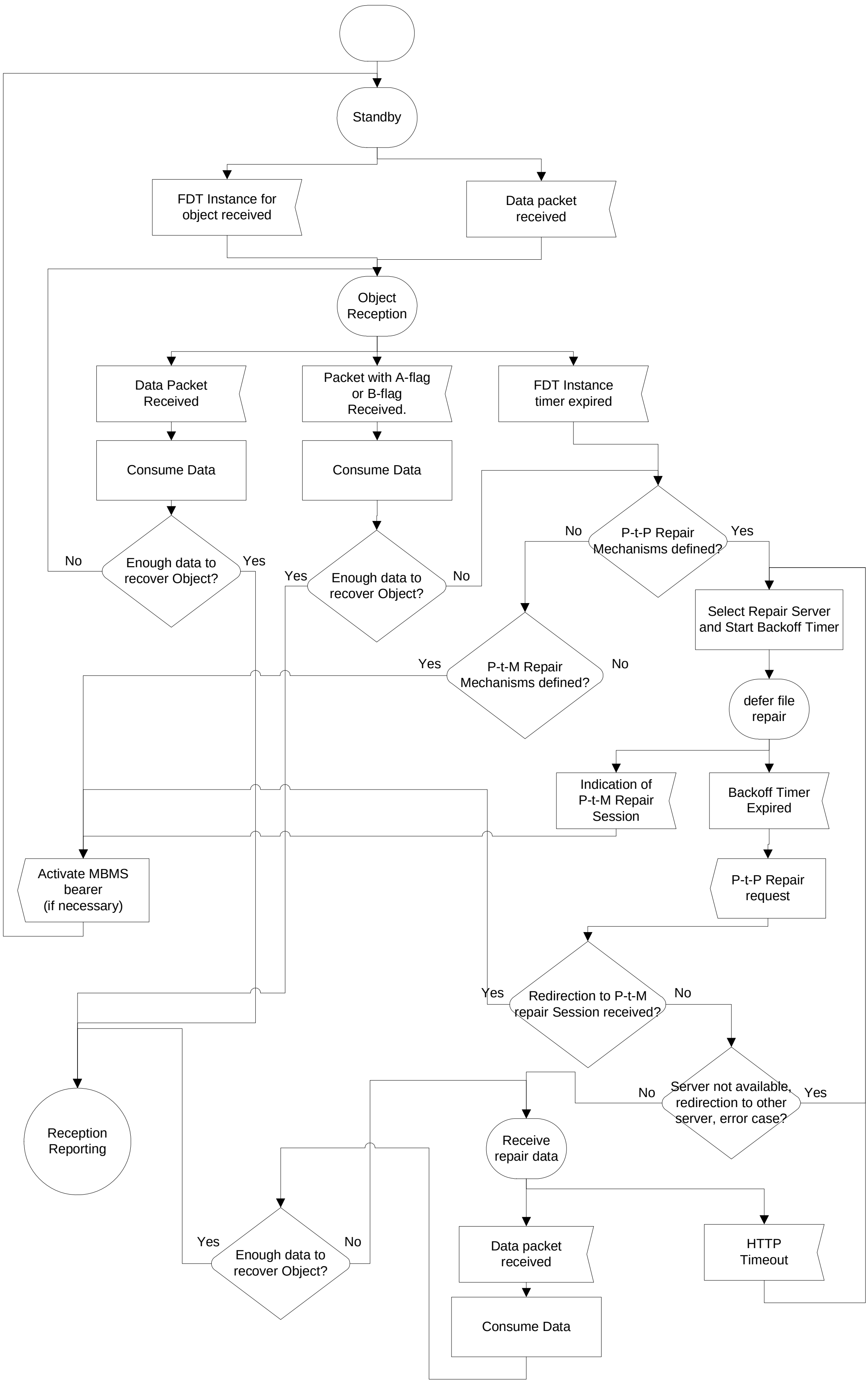
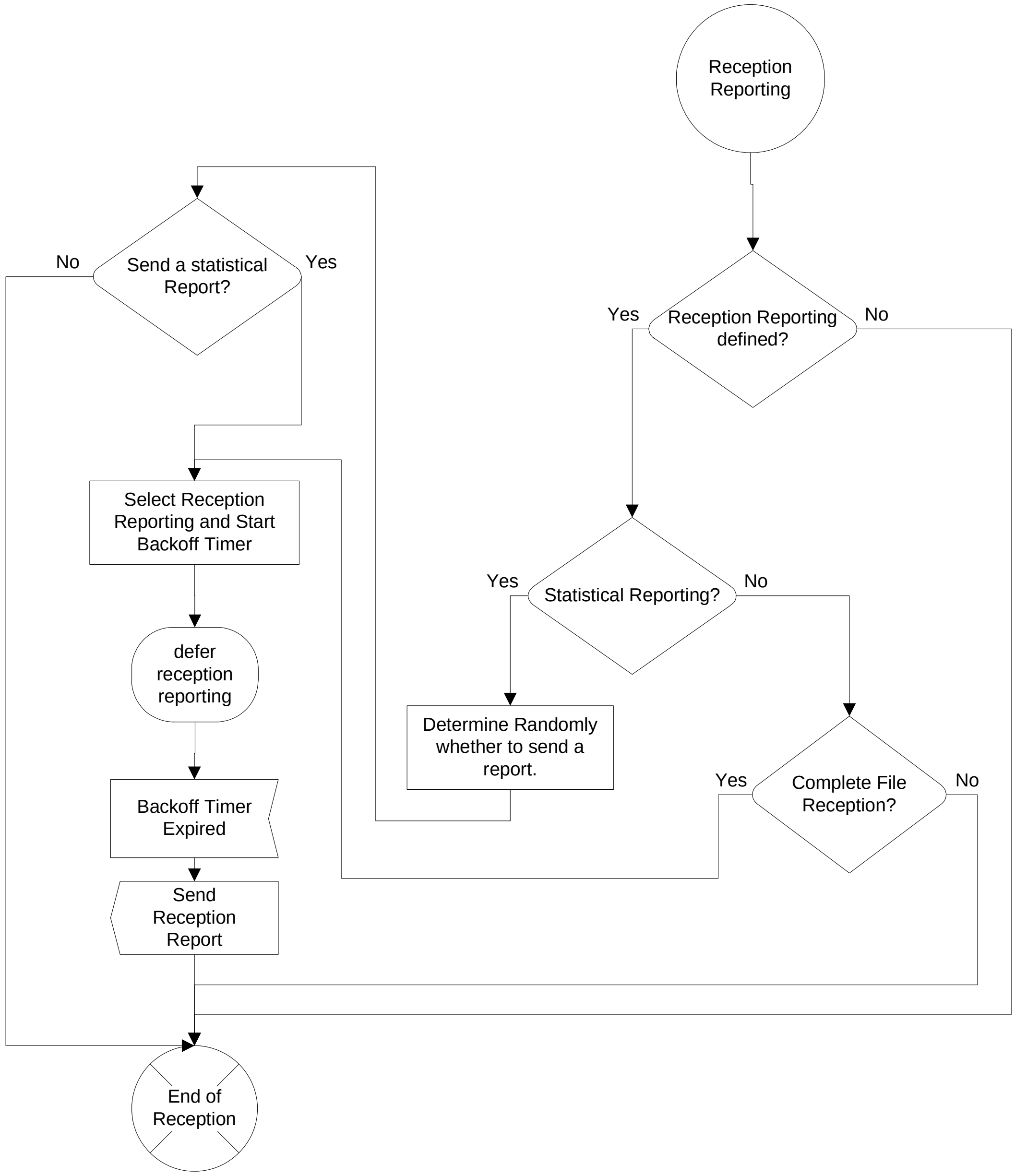


Figure 4: SDL diagram for the file download process of the MBMS UE



NOTE: Statistical Reports for download are only sent, if either a "Close session flag" (see clause 7.2.6) is received or a the "time to" value of the session description (from "t=" in SDP) is reached or a UE decides to exit the session.

Figure 5: SDL diagram for the reception reporting of the file download process of the MBMS UE

6.1.2 Transport of MBMS download data

This clause explains briefly how files are constructed for and transported during a FLUTE session.

The BM-SC takes a file, e.g. a video clip or a still image, which is used as the transport object for FLUTE (see figure 6). The BM-SC constructs source block by breaking the file into contiguous portions of approximately equal size. Each source block is broken into source symbols. One or more encoding symbols are carried as the payload of a FLUTE packet, thus the encoding symbol size must divide the FLUTE packet size. The target FLUTE packet size is configured by the BM-SC and, together with the file size, is used to determine the encoding symbol length. Note that when FEC is used with smaller files it is often necessary to include several symbols in each FLUTE packet. This means that the FLUTE packets cannot be fixed at a specific arbitrary size by the operator, because they must be a multiple of the number of symbols required per packet in size. Based on the transport object length, the encoding symbol length and the maximum source block length, FLUTE calculates the source block structure (i.e., the number of source blocks and their length).

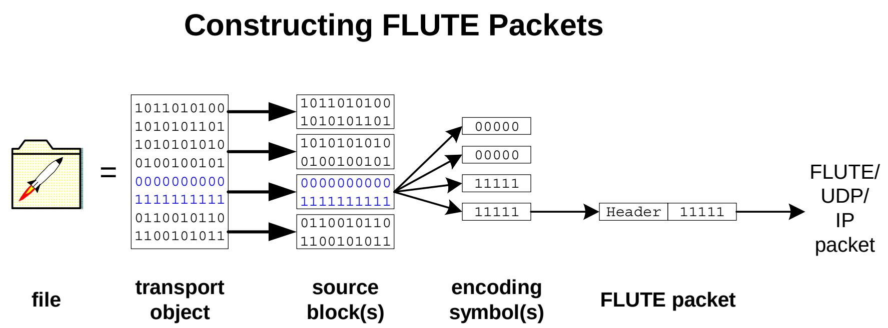


Figure 6: Constructing FLUTE packets

The BM-SC communicates the transport object length, the encoding symbol length and the file size to the receivers within the FLUTE session transmission. Thus the receiver can also calculate the source block structure in advance of receiving a file.

The FLUTE packet is constructed from FLUTE header and payload containing one or more encoding symbols.

The distinction between file and transport object is that the file is the object provided to the BM-SC and played-out or stored at the MBMS UE. Within the scope of FLUTE sessions, content encoding may be used, for instance to compress the file with gzip for delivery. In the presence of FLUTE session content encoding, the file and the transport object will be different binary objects, and in the absence of content encoding the transport object will be identical to the file. Any symbol calculations (including FEC) are performed on transport objects.

EXAMPLE 1:	File size:	1 MB (1 048 576 bytes)
	Target packet size (payload):	500 bytes
	Maximum source symbols:	8 192 (defined by FEC code)
	Minimum source block target:	1 024 (recommended by FEC code)
	Symbol alignment parameter:	4 bytes

Based on the recommendations of 3GPP TS 26.346 [6], clause B.3.4.1, then the above parameters would lead to 1 symbol of 500 bytes per packet and the file would be treated as a single source block with 2 098 symbols.

EXAMPLE 2:	File size:	16 MB (16 777 216 bytes)
	Target packet size (payload):	250 bytes
	Maximum source symbols:	8 192 (defined by FEC code)
	Minimum source block target:	1 024 (recommended by FEC code)
	Symbol alignment parameter:	4 bytes

This would result in 1 symbol of 248 bytes per packet and the file being broken into 9 source blocks, 2 source blocks will have size 7616 symbols and the remaining 7 will have size 7 517 symbols.

EXAMPLE 3:	File size:	256 KB (262 144 bytes)
	Target packet size (payload):	500 bytes
	Maximum source symbols:	8 192 (defined by FEC code)
	Minimum source block target:	1 024 (recommended by FEC code)
	Symbol alignment parameter:	4 bytes

This would result in 2 symbols of 248 bytes each per packet and the file being treated as a single source block of 1 058 symbols.

6.1.3 Repair Symbol Request

In 3GPP TS 26.346 [6], clause 9.3.3, on "Identification of Missing Data from an MBMS Download", it is stated that an MBMS UE is able to identify the ESI values of required source and/or repair symbols that would complete the reconstruction of a source block (of a file) and that the corresponding symbols can be requested from the server. Especially for the case when MBMS FEC scheme is used, the MBMS client should either:

1. identify a minimal set of source symbols that, combined with the already received symbols, allow the MBMS FEC decoder to recover the file; or

2. identify a number, *r*, of symbols such that reception of *r* previously non-received symbols will allow the MBMS FEC decoder to recover the file.

Option 1 (clause 6.1.3.1) is more appropriate if only very few symbols are lost as the signalling of only very few symbols is more efficient. In contrast, option 2 (clause 6.1.3.3) is appropriate in the case where a significant amount of symbols are not available. To minimize the uplink traffic, sending only the initial symbol ESI and the amount of requested consecutive symbols is more appropriate. The decision which option to use is up to the MBMS client.

For both options, example algorithms to determine a suitable set of repair symbols are presented in the following. For option 1 the derivation of a minimum set of source symbols and for option 2 the derivation of a sufficient set of consecutive repair symbols is described based on the initial matrix **A** as constructed in 3GPP TS 26.346 [6], clause C.2.1. In both cases, a maximum Gaussian elimination as described below is performed on **A**. Then, the matrix is virtually extended by those rows which, for option 1, represent the missing source symbols and, for option 2, a set of consecutive repair symbols. Maximum Gaussian elimination is again performed on this new matrix. With appropriate tracking of row and column labels this process allows finding an appropriate set of ESIs for repair of the source block as requested in 3GPP TS 26.346 [6], clause 9.3.3. Note that almost all parts of described algorithms can be included in the regular FEC decoding process as specified in annex C such that the decoding complexity is minimized. Nevertheless, the description of the following algorithms is based on the unmodified matrix **A**.

6.1.3.1 Option 1: Determination of a minimum set of source symbols for repair

The following algorithm operating on matrix **A,** is used to determine a minimum set of source symbols for successful recovery:

1. Assume **c** to be the column index vector of length L with labels such that column exchanges can be tracked, e.g. for all i=0, …, L-1 c[i]=i.

2. Apply the maximum Gaussian elimination process, as described below, to **A**, with tracking of column labels **c**. The process returns matrices **U** and **W** as well as the modified column label vector **c'** of dimension L.

3. Generate an m by L matrix **G'**, whereby m is equal to the number of non-received source symbols as follows. Consider $\mathbf{G_{LT}}$ to be the K by L generator matrix that corresponds to the outer LT encoder of the Raptor code generating the first K encoding symbols according to figure 3GPP TS 26.346 [6], clause B.5.2.5.2-1. The matrix **G**' is constructed from matrix $\mathbf{G_{LT}}$ by deleting the rows corresponding to already received encoding symbols and by exchanging the columns according to **c**'. In addition, the rows in **G'** get assigned the corresponding ESIs and these labels are tracked in **v**'.

4. A new decoding matrix **A'** is constructed in the following way:

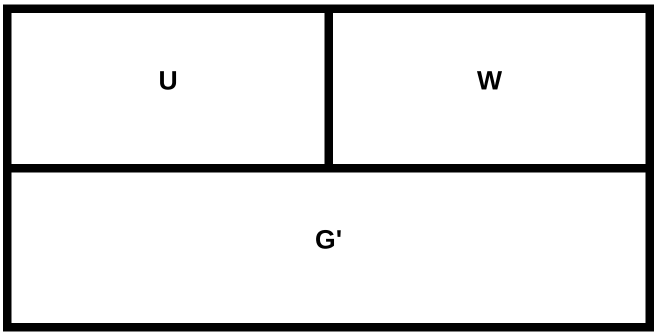


Figure 7: New decoding matrix A'

5. In addition, a row label vector **v** is generated by the vertical stacking of **n** and **v'**, i.e. **v**=[$\mathbf{n}^T$ $\mathbf{v}'^T$]^T whereby **n** represents a vector containing any rank(**U**)=dim(**U**) entries such that any label of **n** can be distinguished from any label in **v**'. In the remainder we assume **n** being an all -1 vector of dimension rank(**U**).

6. Apply the maximum Gaussian elimination process to **A**', as described in clause 6.1.3.3, with tracking of row labels **v**. The process returns the exchanged row labels **v***.

7. The ESIs of the encoding symbols forming a minimum set of repair symbols are obtained by the labels in **v*** which are not -1.

6.1.3.2 Option 2: Determination of a sufficient set of consecutive encoding symbols for repair

The following algorithm operating on matrix **A** is used to determine a sufficient set of consecutive encoding symbols for successful repair:

1. Assume **c** to be the column index vector of length *L* with labels such that column exchanges can be tracked, e.g. for all *i*=0, …, *L*-1 c[*i*]=*i*.

2. Apply the maximum Gaussian elimination process, as described below, to **A**, with tracking of column labels **c**. The process returns matrices **U** and **W** as well as the modified column label vector c' of dimension *L*.

3. Assume that R_{low} is one higher than the largest ESI received and that $R_{\text{high}}= R_{\text{low}}+m-1$. Generate an *m* by *L* matrix G', whereby *m* is equal to or only slightly greater than the minimum number of missing repair symbols *L*-rank(U) by vertically stacking rows that correspond to the outer LT encoder of the Raptor code for all ESI from $R_{\text{low}}, \ldots, R_{\text{high}}$.

4. A new decoding matrix **A**' is constructed according to figure 7.

5. Apply the maximum Gaussian elimination process to **A**' which returns some different **U** and **W** as described in clause 6.1.3.3.

6. If **W** is not the empty matrix, then:

   - Set $R_{\text{high}}= R_{\text{high}}+1$

   - Generate an 1 by L matrix **G**' constructed according to the outer LT encoder of the Raptor code for ESI=R_{high}.

   - Go to 4

7. the ESIs of the encoding symbols forming a minimum set of repair symbols are obtained by requesting all symbols with ESI from $R_{\text{low}}, \ldots, R_{\text{high}}$.

6.1.3.3 Maximum Gaussian elimination

Assume that we have given any matrix **X** as well as possibly a vector of row labels **v** or possibly a column label vector **c** or possibly both. A maximum Gaussian elimination is an algorithm similar to standard Gaussian elimination, in which the main diagonal is extended to its maximum limit, i.e., the number of 1s in the main diagonal after this algorithm has been applied, correspond to the rank of the input matrix **X**, i.e., rank(**X**)..

To be more specific assume that matrix **X** is conceptually divided into 4 parts as shown in figure 8.

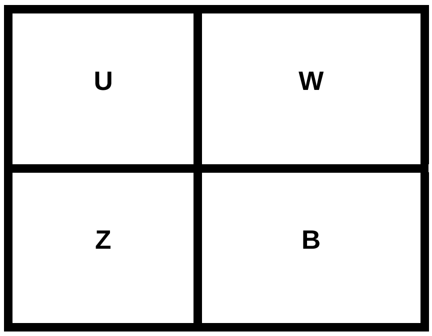


Figure 8: Conceptual division of X into 4 blocks

where **U** is a square-upper-matrix with 1s in the main diagonal and zeros below the main diagonal, **Z** is a zero matrix and **W** and **B** are any arbitrary matrices with appropriate dimensions. Initially, **U**, **W**, and **Z** have dimensions such that **B**=**X**.

In the maximum Gaussian elimination process, the matrix **X** and if present, the row label vector **v** and column label vector **c**, are now processed and modified in the following way:

1. Partition the matrix **X** in **U**, **Z**, **W**, and **B** as shown in figure 8 such that **U** is a square-upper-matrix with 1s in the main diagonal and zeros below the main diagonal, **Z** is a zero matrix and **W** and **B** are any arbitrary matrices with appropriate dimensions.

2. If **B** is the zero matrix, the algorithm stops and returns matrices **U** and **W**, if **v** is present, the first rank(**U**) components of **v** in some vector **v**', and if **c** is present, the entire vector **c**.

3. Otherwise, **B** is transformed using row/column exchange(s) into a matrix with a 1 the in upper-left corner. The following operations on **B** are allowed:

   a) row exchange, accompanied by the exchange of row labels in **v**, if present.

   b) column exchange, accompanied by the exchange of column labels in **c**, if present.

4. by means of row additions, the rows of **B** (except the first row) are eliminated, i.e. 0s are produced in all rows of the first column of **B** except for the first row.

5. the process restarts with the updated matrix **X** in 1.

6.1.4 On choosing the SDU size

This clause provides a simple analysis that suggests how to choose the SDU size. The crux of the analysis is to notice that there is a fundamental tradeoff between two competing factors that suggest how to choose the SDU size. The two factors are the relative length of the SDU header compared to the SDU size, and the SDU loss rate induced by PDU loss. Generally the SDU header size is fixed by other considerations, and the PDU loss rate is determined by characteristics of the underlying network. Given a fixed SDU header size and PDU loss rate, making the SDU larger means the header wastes less space per SDU, whereas making the SDU smaller means the SDU loss induced by PDU loss is smaller. The MBMS FEC code has the property that data (which is a source file in the case of the file download service or a source block in the case of the streaming service) can be recovered with high probability from the reception of any set of encoding packets only slightly greater in length to the data, independent of packet loss amounts or patterns. The loss of SDUs has the same negative impact on wasted bandwidth in terms of delivering the data as the wasted (but necessary) SDU header space. Thus, the goal is to choose an SDU length so that the bandwidth wasted due to headers and SDU loss induced by PDU loss is minimized. (It is important to note that this is not trying to minimize total SDU loss, only the contribution to SDU loss due to PDU loss. Other sources of SDU loss, e.g. cell change loss, cell congestion loss, backbone loss, UE unavailable loss, all independently contribute to bandwidth wastage but do not affect the analysis of how to choose the SDU size based on balancing the header wastage against the PDU loss induced wastage.)

6.1.4.1 Analysis

Let H be the SDU header size (which for example consists of the IP/UDP/FLUTE headers), let B be the PDU size and let p be the PDU loss probability, and all of these parameters are fixed. Let P be the overall SDU size (including headers) that is to be set based on the fixed parameters. Let $h = H/P$ be the fractional header wastage. For example, if $H = 44$ bytes, $P = 440$ bytes then $h = 0.1$. Let q be the SDU loss probability induced by PDU loss. To optimize the value of P, one would want to choose the value of P to make $h + q$ minimal. The contribution [1] derives the value of q as a function of P and B and p. From 3GPP TR 21.905 [1], and using $N \cdot p$ as an upper bound on the probability that N PDUs are lost, an upper bound on the value of q is $(1+P/B)\cdot p$. The actual value of q is at least $(1 - \varepsilon/2)$ times this upper bound when $N \cdot p < \varepsilon$ for $N = \text{ceil}(P/B)+1$, and hereafter the upper bound for q will be used for the actual value of q. To minimize $h + q$ means setting P so as to minimize $H/P + (1+P/B)\cdot p$, and this is minimized by setting $P = \text{sqrt}\{H \cdot B/p\}$. Let $A = \text{sqrt}\{H \cdot p/B\}$. The overall wastage for this value of P is then $h + q = p + 2 \cdot A$.

The values of H and B are determined by the network and link layer protocols used for transport. For example, $H = 44$ bytes (IP/UDP/FLUTE headers) and $B = 640$ bytes for UTRAN or $B = 30$ bytes for GERAN. Note that if Robust Header Compression is applied in the RAN then the value of H in these calculations must be reduced accordingly.

The value of p may or may not be known, but it may be possible to heuristically roughly estimate the average value of p and use this to decide on the value of P.

Suppose for example $H = 44$ bytes, $B = 640$ bytes and $p = 0.1$. The computed value of P is 531 bytes, $h = 0.083$, $q = 0.183$ and thus the overall wastage $h + q = 0.266$. As another example, suppose $H = 44$ bytes, $B = 30$ bytes and $p = 0.01$. The computed value of P is 364 bytes, $h = 0.121$, $q = 0.131$ and thus the overall wastage $h + q = 0.252$.

The penalty for using a value for p to determine P that is larger or smaller than the actual average PDU loss is that there will be wasted bandwidth, either from a header too large if the value of p is estimated higher than the actual PDU loss rate or from too much SDU loss induced by PDU loss if the value of p is estimated lower than the actual PDU loss rate. If for example the actual PDU loss is p and the value used to compute P is $\beta \cdot p$ then the overall wastage will be $p + (\text{sqrt}\{\beta\}+\text{sqrt}\{1/\beta\})\cdot A$ instead of $p + 2 \cdot A$. Since the first term p is the same in both expressions, the ratio of the second terms is an upper bound on the relative wastage of choosing an incorrect p value as a function of β, and this ratio is $(\text{sqrt}\{\beta\}+\text{sqrt}\{1/\beta\})/2$. For example if β is anywhere in the range from ¼ to 4 then the relative wastage is at most a factor of 1.25 higher than it would be if p were estimated precisely, and typically the actual factor (when the first terms are accounted for) is less than 1.25. For example if $H = 44$ bytes, $B = 640$ bytes and $p = 0.1$ and $\beta = ¼$ (meaning that the estimate of the PDU loss used to set the value of P is $\beta \cdot p = 0.025$ whereas the actual PDU loss is $p = 0.1$) then the chosen value of P is 1,062 bytes and the wastage is $0.1 + 2.5 \cdot A$ where $A = 0.083$, and thus the wastage is 0.307. If the correct value of PDU loss were used instead to choose the value of P then P would be set to 531 bytes and the wastage would be $0.1 + 2 \cdot A = 0.266$. Thus, the ratio of the actual to the idea wastage is 1.15.

6.2 Streaming delivery method

The MBMS Streaming Delivery Method aims at continuous transmission of data and the immediate play-out via the display and/or the loudspeaker. Mobile terminals retrieve transmission details like Multicast IP Address, TMGI and the used ports before the MBMS UEs can activate the reception. Upon interaction of the user and when all parameters are known, the UE "tunes in" the transmission and stays until the user decides to leave the transmission. This can happen before the transmission ends.

For Release 6 no dynamic bandwidth adaptation is foreseen. The transmission bandwidth is determined by the BM-SC once and not changed during the life-time of the MBMS streaming session.

7 Usage scenarios

7.1 Service Discovery/Announcement

Below an example for an aggregated service announcement document using multipart MIME type "multipart/related" (RFC2557) is described.

```
MIME-Version: 1.0
Content-Type: multipart/related; boundary=D8119DDD2D264D4480E57277; type=" application/mbms-user-service-description+xml"
Content-Length: <length of the stream>

--D8119DDD2D264D4480E57277

Content-Type: application/mbms-user-service-description+xml

<?xml version="1.0" encoding="UTF-8"?>
<bundleDescription xmlns="urn:3GPP:metadata:2005:MBMS:userServiceDescription"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:3GPP:metadata:2005:MBMS:userServiceDescription">
<userServiceDescription   serviceId="urn:3gpp:1234567890coolcat">
      <name lang="EN">something in english</name>
      <serviceLanguage>EN</serviceLanguage>
      <deliveryMethod
            sessionDescriptionURI="fragmentdir/session1.sdp"/>
      <deliveryMethod
            sessionDescriptionURI="fragmentdir/session2.sdp"
            associatedProcedureDescriptionURI="fragmentdir/procedureX.xml"/>
</userServiceDescription>
</bundleDescription>

--D8119DDD2D264D4480E57277
Content-Type: application/sdp
Content-Location: fragmentdir/session1.sdp

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 IP4 224.1.2.3
t=3034423619 3042462419
b=AS:77
a=source-filter: incl IN IP6 * 2001:210:1:2:240:96FF:FE25:8EC9
a=FEC-declaration:0 encoding-id=1
m=video 4002 RTP/AVP 97 96 100
b=TIAS:62000
b=RR:0
b=RS:600
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=framesize:96 176-144
a=rtpmap: 97 rtp-mbms-fec-tag/90000
a=fmtp:97 opt=96; FEID=129;FIID=12435;FOTI="1SCxWEMNe397m24SwgyRhg=="
a=rtpmap: 100 rtp-mbms-fec-symbols/10000
a=fmtp:100 FEID=129;FIID=12435;FOTI="1SCxWEMNe397m24SwgyRhg=="; min-buffer-time=2600
m=audio 4004 RTP/AVP 99 98 101
b=AS:15
b=RR:0
b=RS:600
a=rtpmap:98 AMR/8000
a=fmtp:98 octet-align=1
a=rtpmap: 99 rtp-mbms-fec-tag/8000
a=fmtp: 99 opt=98;FEID=129;FIID=12435;FOTI="1SCxWEMNe397m24SwgyRhg=="
a=rtpmap: 101 rtp-mbms-fec-symbols/10000
a=fmtp:101 FEID=129;FIID=12435;FOTI="1SCxWEMNe397m24SwgyRhg=="; min-buffer-time=2600
```

[Editor's note: this SDP example needs updating according to the latest SDP definition]

```
--D8119DDD2D264D4480E57277
Content-Type: application/sdp
Content-Location: fragmentdir/session2.sdp
```

```
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 1234
a=FEC-declaration:0 encoding-id=128; instance-id=0
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
a=lang:EN
a=FEC:0
```

[Editor's note: this SDP example needs updating according to the latest SDP definition]

```
--D8119DDD2D264D4480E57277—
Content-Type: application/mbms-associated-procedure-description+xml
Content-Location: fragmentdir/procedureX.xml

<?xml version="1.0" encoding="UTF-8"?>
<associatedProcedureDescription
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.example.com/mbms-associated-descrition.xsd">
  <postFileRepair
    offsetTime="5"
    randomTimePeriod="10">
    <baseURI server="http://mbmsrepair.operator.umts/"/>
    <baseURI server="http://mbmsrepair1.operator.umts/"/>
    <baseURI server="http://mbmsrepair2.operator.umts/"/>
  </postFileRepair>
  <postReceptionReport
    offsetTime="5"
    randomtimePeriod="10"
    reportType="StR-all"
    samplePercentage="100"
    forceTimingIndependence="0">
    <baseURI server="http://mbmsrepair.operator.umts/"/>
  </postReceptionReport>
</associatedProcedureDescription>

--D8119DDD2D264D4480E57277--
```

7.2 MBMS download delivery method

The MBMS Download Services use the functionality of the MBMS Download Delivery Method, although other user services may also build on this delivery method. Users are informed, via the UI, that new information is locally available. However, this is not a constraint or a requirement of an MBMS download user service. User notification and user-interaction with the download process may or may not be part of a download application UI.

After the data transmission phase the MBMS User Service Receiver may request missing data from a "Associated-Delivery Processing Server". Output of the MBMS Receiver is in case of the MBMS Download delivery method a "file". The further processing of the received data and the proper presentation is not in scope of the present document. It is assumed existing user play-out.

7.2.1 Example: Video clip download service use-case

7.2.1.1 Use-case description

The UE has activated a video-clip distribution service. The service starts 5th of August 2005 and ends on 18th of December (half football season). The UE receives every Saturday afternoon (during a time of 4 hours) approximately every 30 minutes a video clip. The BM-SC has the Ipv4 address: 192.168.1.1 and uses the multicast IP address 224.20.20.4 for service distribution.

7.2.1.2 Announced Metadata Fragments

The MBMS User Service is active from the 5th of August until the 18th of December. The MBMS bearers are activated, if there is any video clip to distribute. Video clips are mostly distributed during weekends, seldom during the week.

To activate the video clip service, the client has received the following metadata fragments through a service announcement procedure:

User Service Description Fragment:

```
<?xml version="1.0" encoding="UTF-8"?>
<userServiceDescription
   xmlns="urn:3gpp:metadata:2004:userservicedescription"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:3gpp:metadata:2004:userservicedescription"
   serviceId="urn:VideoClipDistr-1">
   <deliveryMethod
     sessionDescriptionURI="http://example.org/videoclip-distr.sdp"
     associatedProcedureDescriptionURI="http://www.example.com/default-repair-settings.xml"/>
</userServiceDescription>
```

Session Description Fragment:

```
v=0
o=user123 3332188800 3343766400 IN IP4 192.168.1.1
s=VideoClip Distribution Service example
i=More information
t=3332188800 3343766400
a=mbms-mode:broadcast 1234
a=FEC-declaration:0 encoding-id=1
a=source-filter: incl IN IP4 * 192.168.1.1
a=flute-tsi:116
m=application 12345 FLUTE/UDP 0
c=IN IP4 224.20.20.4
b=64
a=lang:DE
a=FEC:0
```

The t= lines includes the session start and session stop information. Session start is 3332188800 (== 5th of August 2005) and session stop is 3343766400 (== 18th of December 2005).

The User service description and the session description fragments are used to activate the MBMS bearer service with multicast IP address 224.20.20.4 and TMGI 1234. The user service uses in this use-case an MBMS broadcast bearer service. The bearer service remains active until the 18th of December.

File Repair Procedure Fragment:

The post delivery file repair procedure is configured with a back-off window of 40 seconds. It was assumed, that a group of 5 000 UEs need 10 kByte each of repair data, assuming that the file repair server can provide repair data with a capacity of 10 Mbps.

```
<?xml version="1.0" encoding="UTF-8"?>
<associatedProcedureDescription
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.example.com/mbms-associated-descrition.xsd">
    <postFileRepair offsetTime="5" maxBackOff="40">
        <serverURI>"http://mbmsrepair.operator.umts/path/repair-service"</serverURI>
        <serverURI>"http://mbmsrepair2.operator.umts/path/repair-service"</serverURI>
    </postFileRepair>
</associatedProcedureDescription>
```

7.2.1.3 Reception of a video clip

The download transmission session starts on the 7th of August 2005, 19:18h in this use-case. The download transmission session contains only one FLUTE file delivery table (FDT), one file and an update of the Associated Delivery Procedure Description (ADPD) instance. The size of the file is 300 kbyte. The file transmission is FEC protected using the MBMS FEC code. A redundancy overhead of 16 % is added to the file transmission. The FDT instance and ADPD instance are not FEC protected because they are too small in size to benefit from FEC protection. To ensure their reliable delivery, FDT instance and ADPD instance may be transmitted repeatedly.

Note that ADPD has critical information relevant to point-to-point file repair and delivery verification reports. If the ADPD is sent in-band alone and is lost during in-band FLUTE transmission, then point-to-point file repair is not possible. Hence it is recommended to provide a default ADPD instance in User Service Description. The most recently delivered configuration file takes priority. ADPD received in-band with the download session, overwrites earlier received parameters. In this example, the update of the Associated Procedure Description instance contains the URI of an additional repair server.

```
<?xml version="1.0" encoding="UTF-8"?>
<associatedProcedureDescription
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.example.com/mbms-associated-descrition.xsd">
    <postFileRepair offsetTime="5" maxBackOff="40">
        <serviceURI>http://mbmsrepair.operator.umts/path/repair-service</serviceURI>
        <serviceURI>http://mbmsrepair2.operator.umts/path/repair-service</serviceURI>
        <serviceURI>http://mbmsrepair3.operator.umts/path/repair-service</serviceURI>
    </postFileRepair>
</associatedProcedureDescription>
```

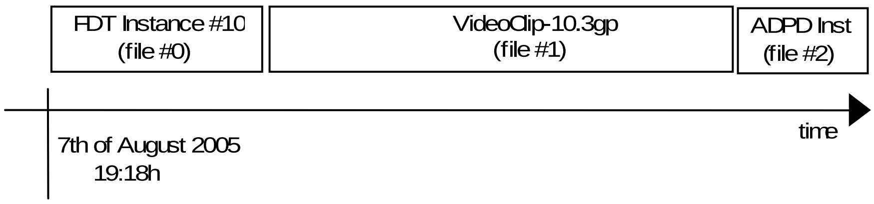


Figure 9: Delivery of video clip

The FLUTE File delivery table is transmitted using the Transport Object ID number 0. The BM-SC starts sending the FLUTE FDT packets at 19:18h.

The TSI and the TOI fields are set to 16bit respectively. The TSI value for the session is 116. TOI value for FLUTE FDT packets is 0. The length of the FDT Instance object is 681 byte. The Compact No Code FEC is used for FDT transmission. The symbol length is set to 512 byte. Maximum source block length is set to 1000.

The FLUTE packet headers, which are used to forward the FDT are depicted below. All values in decimal:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | 0001  | 00| 00|0| 00|1|0|0|0|0|      8        |       0       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                           0   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                          116 |                          0     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   HET = 64    |       4       |                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
 |                               681                             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                0              |                512            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                            1000                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   HET = 192   |    1   |               10                     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |      Source Block Number      |      Encoding Symbol ID       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |             Encoding Symbol(s) for FDT Instance               |
 |                             ...                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: FLUTE FDT packet header

The FLUTE File Delivery Table instance for this use-case is depicted below. The FDT instance expires on the 7th of August 2005, 19:58:46h (3332430520 in NTP format).

```
<?xml version="1.0" encoding="UTF-8"?>
<FDT-Instance xmlns="http://www.example.com/flute"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.example.com/flute"
Expires="3332430526">

    <File TOI="1" Content-Location="http://www.example.com/bundesliga/VideoClip-10.3gp"
Content-Type="video/3gpp"
           FEC-OTI-FEC-Encoding-ID="1"
Transfer-Length="307200"
FEC-OTI-Encoding-Symbol-Length="256"
FEC-OTI-Scheme-Specific-Info="MDAwMTAyMDQ="/>
<File TOI="2" Content-Location="http://www.example.com/bundesliga/adpd-inst.xml"
    Content-Type="application/mbms associated-procedure-description+xml" FEC-OTI-FEC-Encoding-ID="0"
    Transfer-Length="520" FEC-OTI-Maximum-Source-Block-Length="2" FEC-OTI-Encoding-Symbol-
Length="512"/>
</FDT-Instance>
```

The FEC Encoding ID field identifies the MBMS FEC scheme (i.e. Raptor Code). The value "1" is assigned by IETF to the Raptor Code.

The value of the FEC scheme specific info is base64 Binary encoded. The base 64 value "MDAwMTAyMDQ=" decodes to 0x10204hex (Z = 1; N = 2; A = 4).

The FDT expiration time (i.e. "expire" attribute) is an unsigned integer representing the time in seconds relative to 0 hours 1 January 1900 (i.e. the 32 most significant bits of a 64 bit Network Time Protocol (NTP) time value). If a UE is not time synchronized with the mobile network, then the UE must calculate an expiration time, which is relative to the reception of the FLUTE FDT packet.

7.2.1.4 File Repair procedure

The file correction use case is based on the file reception use-case described in clause 7.2.1.3.

The post delivery file repair procedure (i.e. one possible associated delivery procedure) is configured with a back-off window of 40 seconds. It was assumed, that a group of 5 000 UEs would need 10 kByte each of repair data, assuming that the file repair server can provide repair data with a capacity of 10 Mbps.

```
<?xml version="1.0" encoding="UTF-8"?>
<associatedProcedureDescription
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.example.com/mbms-associated-descrition.xsd">
    <postFileRepair offsetTime="5" maxBackOff="40">
        <serviceURI>http://mbmsrepair.operator.umts/path/repair-service</serviceURI>
        <serviceURI>http://mbmsrepair2.operator.umts/path/repair-service</serviceURI>
        <serviceURI>http://mbmsrepair3.operator.umts/path/repair-service</serviceURI>
    </postFileRepair>
</associatedProcedureDescription>
```

The following table is created assuming the recommended MBMS FEC parameter configurations.

File-Size	300 kbyte
FEC Overhead	16%
Z (Number of Source Blocks)	1
N (Number of sub-blocks)	2
A (Alignment factor)	4
P (Payload size)	512 byte
G (Symbols per packet)	2
T (Symbol Size)	256 byte
Generated Source Packets	600
Generated Repair Packets	96

UE_A:

It is assumed that UE_A switched from a very good coverage to almost no coverage 20 seconds before the transmission ended. It is assumed for UE_A, that a user enters a subterranean garage or a sub-way station while receiving an MBMS download transmission session.

The UE_A misses the last half of the MBMS transmission, thus packet 348 until packet 696 are lost. The file transmission was organized such that the repair packets are transmitted after the source packets. Thus, a part of the source and all repair packets are lost. Since there are two symbols per packet, the corresponding start ESI of the lost packets is 696.

After the FDT instance has expired (thus 46 seconds after receiving the FLUTE FDT packet), the UE_A starts the file repair procedure with calculating the procedure back-off time.

$$\text{Backoff-time} = \text{rnd}() \times 40\ \text{s} + 5\ \text{s}$$

The UE_A selects randomly one server URI from the Associated delivery procedure. In this use-case the UE chooses the second server URI: http://mbmsrepair2.operator.umts.

For this use-case the rnd() function has returned "0.275" as result. Thus, the UE defers the opening of the TCP connection to the host mbmsrepair2.operator.umts by 16 seconds. As soon as the TCP connection is established, the UE sends the following HTTP GET request to the file repair server.

The File Repair request line:

GET http://mbmsrepair2.operator.umts/path/repair-service?fileURI=www.example.com/bundesliga/VideoClip-10.3gp&SBN=1;ESI=696-1198 HTTP/1.1

Since the MBMS UE has successfully received the first part of the source block, it needs to request only the remaining part of the source block to completely recover the file. The MBMS FEC decoder need not be invoked in this example.

The UE_A gets the remaining parts of the file as response.

UE_B:

UE_B switched from a very good coverage to almost no coverage 20 seconds before the transmission ended and re-entered good coverage 15 seconds later.

The UE_B thus misses packets 348 to packet 607 (inclusive). The file transmission was organized such that the repair packets are transmitted after the source packets. Thus, a part of the source is lost (i.e. 251 source packets), but most repair packets have been received (i.e. 88). The total number of received packets is 435 and since there are two symbols per packet, then 870 symbols have been received.

After the FDT instance has expired (thus 46 seconds after receiving the FLUTE FDT packet), the UE_A starts the file repair procedure with calculating the procedure back-off time.

$$\text{Backoff-time} = \text{rnd}() \times 40\ \text{s} + 5\ \text{s}$$

The UE_B selects randomly one server URI from the Associated delivery procedure. In this use-case the UE chooses the second server URI: http://mbmsrepair2.operator.umts

For this use-case the rnd() function has returned "0.275" as result. Thus, the UE defers the opening of the TCP connection to the host mbmsrepair2.operator.umts by 16 seconds. As soon as the TCP connection is established, the UE sends the following HTTP GET request to the file repair server.

A minimum of 330 additional symbols are required to ensure successful decoding of the file. The UE should use one of the algorithms defined in clause 6.1.3 to determine a range of ESIs that will result in successful reception. In this case, the UE determines that an additional 12 symbols (1 %) will be sufficient to ensure recovery. The UE thus requests an additional 342 symbols with ESIs of 1 392 (one greater than the highest ESI received) or above.

The File Repair request line:

GET http:// mbmsrepair2.operator.umts/path/repair-service?fileURI=www.example.com/bundesliga/VideoClip-10.3gp&SBN=1;ESI=1392+342 HTTP/1.1

UE_B receives the repair symbols in response and can thus decode the file. Note that the UE is not permitted to request all missing source symbols, since this would result in an unnecessary load on the repair server (in this case 504 symbols).

7.2.2 The usage of the MBMS session Identity in file download delivery

7.2.2.1 Introduction

The MBMS Identity field can be used by the BM-SC and terminal to establish an n-to-m mapping between the files of a file download sessions and their corresponding MBMS bearer sessions over which the files are transported.

According to 3GPP TS 23.246 [4], the UEs are notified about the starting (or ongoing) data transmission through MBMS notification procedure (see the example of GERAN in figure 11). Both the TMGI and an optional Session ID are paged to the terminals, independent of their current state (idle or connected), to inform them about the starting data transmission. The UEs use the TMGI and the Session ID to decide whether they are interested in the MBMS session or not. In case of session repetitions, the BM-SC assigns the same TMGI and Session ID to the MBMS session. This allows the UEs to recognize that the session is repeated and decide not to receive the data in case they already received it correctly.

The usage of the Session ID field is optional. Both UE and BM-SC may decide to ignore the Session ID field. The UE will then assume that the MBMS session is a new session and make its decision independently. The BM-SC may decide not to use the Session ID field, in which case it will not signal it to the UEs within the FDT nor during the notification.

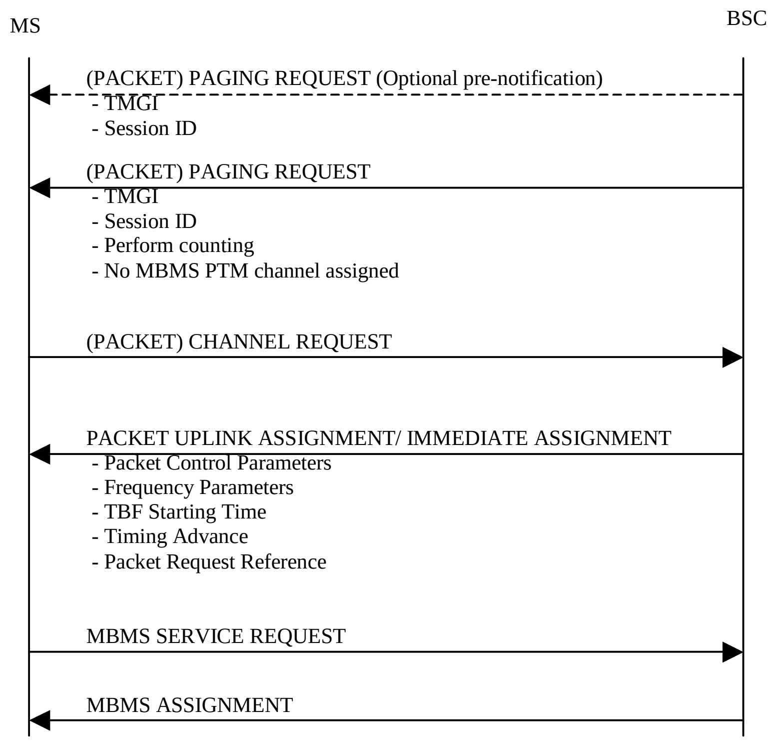


Figure 11: Notification process in the GERAN

The BM-SC uses the FDT of a file download session to indicate for a file or a file group the Session ID value of the MBMS session to carry the file(s). The UE establishes a table to save the mappings between the Session IDs and the files. Each record has a validity time, which is signalled by the "MBMS-Session-Identity-Expiry" in the FDT-Instance. This expiry time applies to all MBMS Session Identity values declared in the same FDT Instance.

In the following we give some examples of the usage of the MBMS Session Identity.

7.2.2.2 Example 1: Usage of the Session ID to detect repetition

Figure 12 depicts the scenario where the terminal receives a notification about a pending MBMS bearer session with session ID value 5. The UE is not aware of the content of the new bearer session, so that it assumes that it is interested in that session. Hence, the UE replies to the notification message to declare to the network that it will receive the data over that MBMS bearer session (which is identified by the TMGI and the Session ID).

Later on, during the download delivery session, the BM-SC decides to retransmit the same files and then uses the same Session ID. In the FDT of the repeated transmission, the same files are declared to belong to the same bearer session. The UE is notified about the new session using the paging information. The UE detects that it has already seen and received the files of that MBMS bearer session correctly. Hence, it decides not to reply to the paging messages and will not receive the corresponding data.

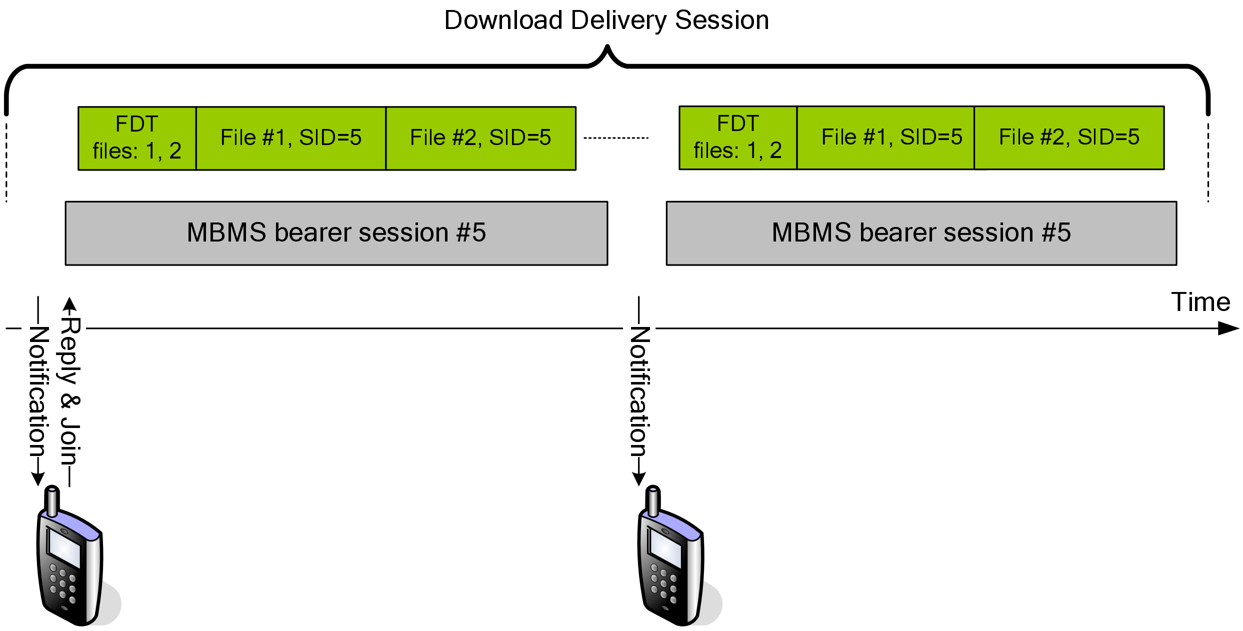


Figure 12: Example of the usage of the Session ID

7.2.2.3 Example 2: Declaration of files of a different MBMS bearer session in the FDT

Figure 13 depicts the scenario where the BM-SC sends an FDT Instance that declares files, which are sent over a different MBMS bearer session. Although file #2 itself is not sent over MBMS bearer session #3, it is declared in the FDT Instance. The UE will detect this and will know the content (or a subset of it) of a forthcoming MBMS bearer session with the Session ID 5.

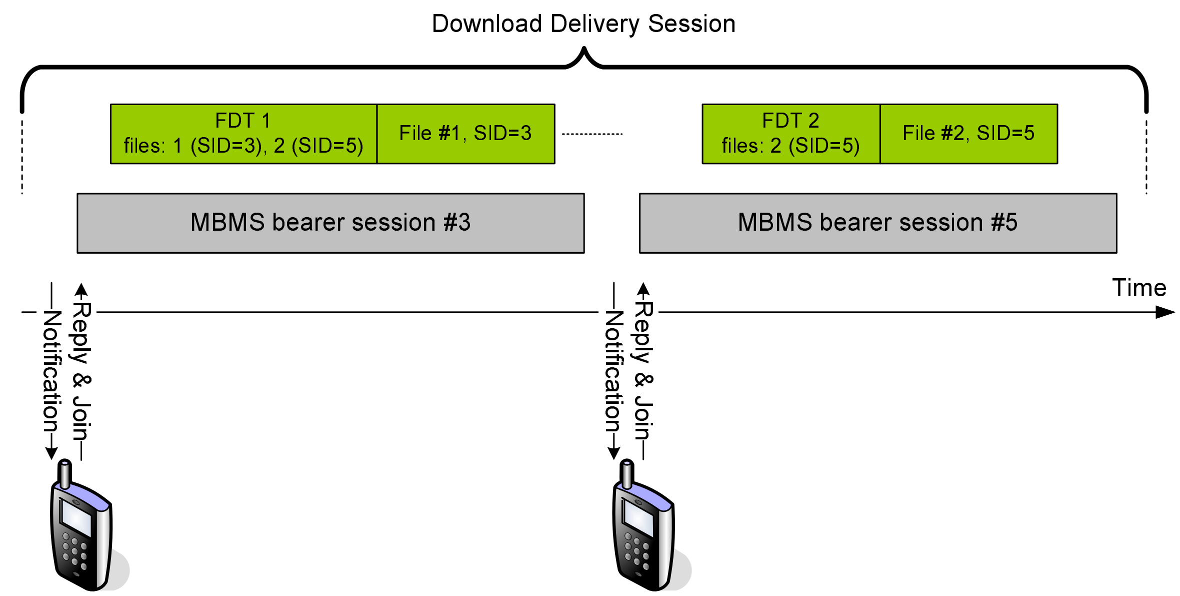


Figure 13: Example of an FDT that declares files of a different FDT

7.2.2.4 Example 3: Declaration of files of an MBMS bearer session in several FDT Instances

Figure 14, the example of splitting the declaration of files carried over a given MBMS bearer service is split over several FDT Instances. It also shows that if one of the FDT Instances and the corresponding files are lost, the UE will not be aware about the full content of the MBMS bearer session. This may lead to a wrong behaviour of the UE, in that it may decided not to receive session repetitions, assuming that it correctly received all files.

It is hence recommended that a single FDT Instance is used to declare all files to be carried over the MBMS bearer session.

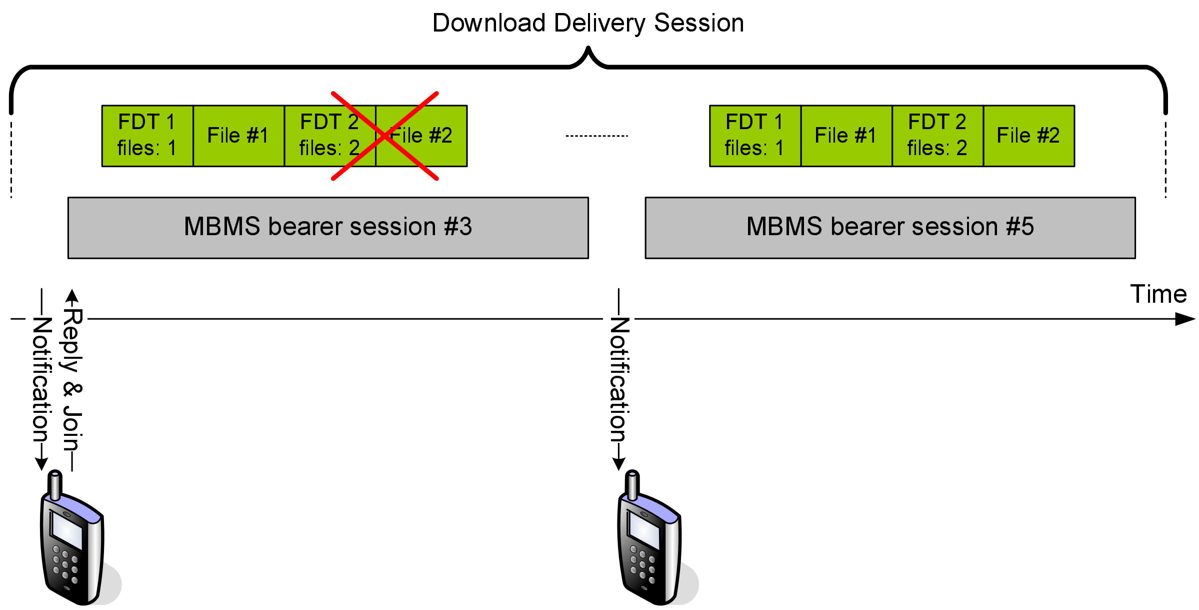


Figure 14: Example of several FDT Instances declaring files for a given MBMS bearer session

7.3 MBMS streaming delivery method

The MBMS Streaming receiver application may be started upon user interaction. In the case of immediate play-out upon reception, the user would be fully aware of the MBMS Streaming reception. The MBMS Streaming User Service is likely a fully integrated application (such as PSS).

The "MBMS User Service Transmitters" sends in case of the MBMS Streaming User service a number of continuous RTP-streams via the MBMS Bearer Service to the "MBMS User Service Receivers". The number of streams per MBMS User Service depends on the multimedia components of the Service.

Output of the MBMS User Service Receiver is in case of the MBMS Streaming User Service a number of continuous media streams (e.g. audio and video stream). The required codecs for reception of MBMS Streaming User Service are provided during the MBMS Service Discovery phase.

The "MBMS User Service Transmitter" starts and stops sending the RTP-streams according to the "Transmission Controller". Users do not have a direct influence on the transmission duration or on the quality of the transmitted media.

8 Codecs and formats

The H.264 (AVC) decoder in an MBMS client should infer missing macroblocks in a coded picture. When gaps_in_frame_num_value_allowed_flag is equal to 0 in the active sequence parameter set and the value of frame_num syntax element in the current picture is invalid according to clause 7.4.3 of ITU-T Recommendation H.264 [17], the H.264 (AVC) decoder in an MBMS client should infer an unintentional reference picture loss. When gaps_in_frame_num_value_allowed_flag is equal to 1 in the active sequence parameter set and any value of frame_num pertaining to "non-existing" pictures is referred to in the inter prediction process, is referred to in the reordering commands for reference picture lists for short-term pictures (clause 8.2.4.3.1 of ITU-T Recommendation H.264 [17]), or is referred to in the assignment process of a LongTermFrameIdx to a short-term picture (clause 8.2.5.4.3 of ITU-T Recommendation H.264 [17]), the H.264 (AVC) decoder in an MBMS client should infer an unintentional reference picture loss. The H.264 (AVC) decoder in an MBMS client should infer a syntax violation, when decoding causes any syntax or semantics violation other than specified above.

When the H.264 (AVC) decoder in an MBMS client infers missing macroblocks in a non-reference picture, it should continue decoding no later than from the next access unit in decoding order. When the H.264 (AVC) decoder in an MBMS client infers an unintentional reference picture loss or missing macroblocks in a reference picture, it may or may not continue decoding "immediately", i.e. from the NAL unit in which the loss is inferred. If the H.264 (AVC) decoder in an MBMS client continues decoding "immediately", it should be aware that such a stream may contain references to macroblocks or pictures that are not available in the decoded picture buffer. The H.264 (AVC) decoder in an MBMS client should continue decoding after an unintentional reference picture loss, missing macroblocks in a reference picture or a syntax violation no later than it receives the next IDR access unit or the next recovery point SEI message, whichever is earlier in decoding order.

NOTE: When the H.264 (AVC) decoder in an MBMS client continues decoding "immediately" after an inferred reference picture loss or missing macroblocks in a reference picture, it does not have to parse recovery point SEI messages to comply with the paragraph above.

Encoding and Packetization Recommendations

The following recommendations are given to MBMS encoders and packetizers to optimize the tradeoff between loss rates and throughput for MBMS video services:

- The encoder/packetizer **should** choose a suitable IP packet size for the loss regime and other network characteristics.
- When H.264/AVC is in use and FEC is in use, the encoder **may** produce NAL units larger than the suitable IP/RTP packet size.
- When H.264/AVC is in use and in any case when larger NAL units are produced by the encoder, the packetizer **should** use NAL Unit fragmentation as specified in RFC 3984 [18], section 5.8, to adapt the RTP packet size to the network, and not produce large RTP packets (which would imply either IP fragmentation or IP packets which use several transmission units, both of which are undesirable).

NOTE: When H.263 is in use, follow-on packets **should** be used for similar purposes as fragmentation units.

Annex A: MBMS Forward Error Correction performance

This annex provides details of the performance of the MBMS Forward Error Correction code which can be used in dimensioning MBMS services.

A.1 Theoretical performance

The theoretical performance of the MBMS FEC code depends on the amount of data received by the terminal. If less data has been received than the size of the file or block to which FEC protection has been applied then recovery of the original data is obviously impossible. If an amount of data has been received which is not less than the size of the file or block then recovery may be possible.

As a general rule, if at least one source packet has been lost and if an amount of data has been received which is 1 % greater than the source size, then the probability of recovery is around 99.9 % or greater. If an amount of data has been received which is 2 % greater than the source size then the probability of recovery is 99.9999 % or greater.

In practice, the recovery probabilities are often higher than these figures, in particular when the number of FEC encoding symbols in the block is large.

A.2 Simulation results

This clause provides results of simulations which were carried out during the development of 3GPP TS 26.346 [5]. These results may be used as guidelines for the provisioning of MBMS services. In general, provisioning should be based on first determining the available resources for the MBMS service and the coverage required. The service should then be dimensioned to provide acceptable quality of service at the edge of the coverage area.

It should be noted that as discussed in clause 6.1.4, the IP packet loss rate may be quite sensitive to the chosen SDU size. Therefore care should be taken when using the figures presented here with SDU sizes, or other configuration parameters, which differ significantly from those used for the simulations.

A.2.1 Simulation conditions and assumptions (UTRAN)

UTRAN Download		
	Bearer rates	64 kbit/s, 128 kbit/s, 256 kbit/s
	RLC PDU size	640 bytes, 1 280 bytes, 1 280 bytes respectively
	RLC BLER	1%, 5%, 10%, 15%, 20%, 30%
	RLC block loss pattern	Independent random loss
	Number of trials	At least 10,000 for files ≤ 512 KB, 3,000 for 3 072 KB
	File sizes	50 KB, 512 KB, 3 072 KB
	FLUTE payload size	456 bytes
	ROHC	No
	IPv4/UDP header	28 bytes
	FLUTE header	16 bytes
	FEC overhead	Varied in steps of X packets, where X=ceil(0.005N) and N is the number of packets containing source data
UTRAN Streaming		
	Bearer rates	64 kbit/s, 128 kbit/s and 256 kbit/s
	RLC PDU size	640 bytes (for 64 kbit/s bearer) 1280 bytes (for 128 kbit/s bearer) 1280 bytes (for 256 kbit/s bearer)
	RLC BLER	1 %, 5 %, 10 %, 15 %, 20 %, 30 %
	RLC block loss pattern	Independent random loss
	Simulation duration	24 hours
	Media rates	Varied by steps of 1 % of bearer rate, assuming only a single media stream (see note 1)
	FEC overhead	Varied to sum FEC and Media to equal bearer rate
	Source packet RTP payload size	64 kbit/s: 456 bytes 128 kbit/s: 456 bytes 256 kbit/s: 768 bytes
	Repair packet RTP payload size	Minimum value supported by the FEC code which is not less than 470 (for 64 kbit/s and 128 kbit/s) and 782 (for 256 kbit/s) - (see note 2)
	Protection period	5 s, 20 s
	ROHC	No
	IPv4/UDP/RTP header	40
NOTE 1: In practice, multiple media streams may be carried within a single MBMS bearer. However, only a single media stream is considered for FEC simulation purposes for simplicity. NOTE 2: The last repair packet of a block may be shorter if supported by the FEC code in order to fit within the protection period.		

A.2.2 Simulation conditions and assumptions (GERAN)

GERAN Download		
	Bearer rates	28.8 kbit/s, 59.2 kbit/s, 118.4 kbit/s
	RLC PDU size	36 bytes, 74 bytes, 74 bytes respectively
	RLC BLER	for 28.8 kbit/s 0.1 % for 59.2kbit/s : 0.5 % for 118 kbit/s: 1 %, 10 %
	RLC block loss pattern	Independent random loss
	Number of trials	At least 10,000 for files ≤ 512 KB, 3,000 for 3 072 KB
	File sizes	50 KB, 512 KB, 3 072 KB
	FLUTE payload size	456 bytes, (for 10 % BLER at 118.4 kbps also simulate a case with 146 bytes payload)
	ROHC	No
	SNDCP/LLC/IPv4/UDP header	38 bytes
	FLUTE header	16 bytes
	FEC overhead	Varied in steps of X packets, where X=ceil(0.005N) and N is the number of packets containing source data
GERAN Streaming		
	Bearer rates	28.8 kbit/s, 59.2 kbit/s and 118.2 kbit/s
	RLC PDU size	36 bytes, 74 bytes, 74 bytes, respectively
	RLC BLER	for 28.8 kbit/s 0.1 % for 59.2kbit/s : 0.5 % for 118 kbit/s: 1 %, 10 %
	RLC block loss pattern	Independent random loss
	Simulation duration	24 hours
	Media rates	Varied by steps of 1 % of bearer rate, assuming only a single media stream (see note 1)
	FEC overhead	Varied to sum FEC and Media to equal bearer rate
	Source packet RTP payload size	456 bytes (for 10 % BLER at 118.4 kbps also simulate a case with 146 bytes payload)
	Repair packet RTP payload size	Minimum value supported by the FEC code which is not less than 470 bytes (for 10 % BLER at 118.4 kbps also simulate a case with 160 bytes payload) - (see note 2)
	Protection period	5 s, 20 s
	ROHC	No
	SNDCP/LLC/IPv4/UDP/RTP header	50
NOTE 1: In practice, multiple media streams may be carried within a single MBMS bearer. However, only a single media stream is considered for FEC simulation purposes for simplicity. NOTE 2: The last repair packet of a block may be shorter if supported by the FEC code in order to fit within the protection period.		

A.2.3 Simulation results: UTRAN download

Table A.1: FEC Overhead required for 99 % probability of recovery at specific BLER points 64 kbit/s

Error rates	Power required (G=-3dB - note 1)	Power required (G=-6dB - note 2)	File size	Ideal (note 3)	MBMS FEC
Low (1 % BLER)	2.0 %	4.5 %	Small (50 KB)	7.0 %	8.0 %
			Medium (512 KB)	3.3 %	3.6 %
			Large (3 072 KB)	2.4 %	2.6 %
Medium (5 % BLER)	1.8 %	3.9 %	Small (50 KB)	21.8 %	22 %
			Medium (512 KB)	13.0 %	13.4 %
			Large (3 072 KB)	11.0 %	11.2 %
High (10 % BLER)	1.7 %	3.7 %	Small (50 KB)	39.0 %	39.0 %
			Medium (512 KB)	25.8 %	26.0 %
			Large (3 072 KB)	22.6 %	22.8 %
15 % BLER	n/a	n/a	Small (50 KB)	56.0 %	56.0 %
			Medium (512 KB)	40.5 %	41.0 %
			Large (3 072 KB)	36.0 %	37.0 %
20 % BLER	n/a	n/a	Small (50 KB)	76.0 %	76.0 %
			Medium (512 KB)	57.0 %	57.0 %
			Large (3 072 KB)	52.0 %	52.0 %
30 % BLER	n/a	n/a	Small (50 KB)	130.0 %	130.0 %
			Medium (512 KB)	100.0 %	100.0 %
			Large (3 072 KB)	92.0 %	92.0 %
NOTE 1: This corresponds to ~90 % of users assuming uniform distribution of users. NOTE 2: This corresponds to ~99 % of users assuming uniform distribution of users. NOTE 3: This column provides the best possible performance which could be achieved with a theoretical "ideal" forward error correction code.					

Table A.2: FEC Overhead required for 99 % probability of recovery at specific BLER points 128 kbit/s and 256 kbit/s

Error rates	Power required (G=-3dB - note 1) 128 kbit/s 256 kbit/s	Power required (G=-6dB - note 2) 128 kbit/s 256 kbit/s	File size	Ideal (note 3)	MBMS FEC
Low (1 % BLER)	4.0 % 7.9 %	8.9 % 19.3 %	Small (50 KB)	7.5 %	8.0 %
			Medium (512 KB)	3.1 %	3.4 %
			Large (3 072 KB)	2.1 %	2.2 %
Medium (5 % BLER)	3.7 % 7.1 %	7.8 % 16.0 %	Small (50 KB)	20.0 %	21.0 %
			Medium (512 KB)	11.2 %	11.4 %
			Large (3 072 KB)	8.8 %	9.0 %
High (10 % BLER)	3.4 % 6.8 %	7.2 % 14.8 %	Small (50 KB)	35.0 %	35.0 %
			Medium (512 KB)	21.5 %	21.5 %
			Large (3 072 KB)	17.8 %	18.1 %
15 % BLER	n/a	n/a	Small (50 KB)	50.0 %	50.0 %
			Medium (512 KB)	32.0 %	32.1 %
			Large (3 072 KB)	28.0 %	28.1 %
20 % BLER	n/a	n/a	Small (50 KB)	66.0 %	66.0 %
			Medium (512 KB)	44.6 %	45.0 %
			Large (3 072 KB)	38.0 %	38.2 %
30 % BLER	n/a	n/a	Small (50 KB)	106.0 %	106.0 %
			Medium (512 KB)	72.0 %	72.0 %
			Large (3 072 KB)	66.8 %	67.0 %
NOTE 1: This corresponds to ~90 % of users assuming uniform distribution of users. NOTE 2: This corresponds to ~99 % of users assuming uniform distribution of users. NOTE 3: This column provides the best possible performance which could be achieved with a theoretical "ideal" forward error correction code.					

A.2.4 Simulation results UTRAN streaming

Table A.3: Maximum supported Media Rate (kbit/s) for Mean Time Between FEC Block Loss of 1 hour

Error rates	Bearer rate	Ideal (see note)	MBMS FEC
Low (1 % BLER)	Low (64 kbit/s)	5 s: 56.8 20 s: 60.6	5 s: 55.8 20 s: 60.4
	Medium (128 kbit/s)	5 s: 116.3 20s: 122.6	5 s: 115.5 20 s: 122.4
	High (256 kbit/s)	5 s: 237.4 20 s: 246.4	5 s: 236.2 20 s: 245.7
Medium (5 % BLER)	Low (64 kbit/s)	5 s: 47.4 20 s: 54.2	5 s: 46.6 20 s: 53.6
	Medium (128 kbit/s)	5 s: 102.2 20 s: 112.5	5 s: 100.8 20 s: 111.8
	High (256 kbit/s)	5 s: 228.0 20 s: 224.5	5 s: 227.0 20 s: 224.0
High (10 % BLER)	Low (64 kbit/s)	5 s: 39.5 20 s: 47.5	5 s: 38.5 20 s: 47.2
	Medium (128 kbit/s)	5 s: 88.5 20 s: 101.8	5 s: 87.5 20 s: 101.2
	High (256 kbit/s)	5 s: 182.0 20 s: 201.5	5 s: 179.5 20 s: 200.5
NOTE: This column provides the best possible performance which could be achieved with a theoretical "ideal" forward error correction code.			

A.2.5 Simulation results GERAN download

Table A.4: FEC Overhead required for 99 % probability of recovery at specific GERAN operation points

Operation Points	File size	Ideal (see note)	MBMS FEC
Low Bitrate (CS3) 0.1% BLER 28.8 kbit/s	Small (50 KB)	4.7 %	5.3 %
	Medium (512 KB)	2.4 %	2.7 %
	Large (3 072 KB)	1.8 %	2.1 %
Medium Bitrate (MCS-6) 0.5% BLER 59.2 kbit/s	Small (50 KB)	9.4 %	9.7 %
	Medium (512 KB)	5.6 %	5.9 %
	Large (3 072 KB)	4.6 %	4.7 %
High Bitrate (MCS-9) 1% BLER 118.4 kbit/s	Small (50 KB)	16.0 %	16.0 %
	Medium (512 KB)	10.6 %	10.8 %
	Large (3 072 KB)	9.2 %	9.3 %
High Bitrate and High Error Rate (MCS-6) 10% BLER, 146 byte packet payload 118.4 kbit/s	Small (50 KB)	60 %	60 %
	Medium (512 KB)	51.5 %	51.5 %
	Large (3 072 KB)	49.2 %	52.0 %
NOTE: This column provides the best possible performance which could be achieved with a theoretical "ideal" forward error correction code.			

NOTE: 440 byte payloads, according to Raptor specification recommendations.

A.2.6 Simulation results GERAN streaming

Table A.5: Maximum supported Media Rate for Mean Time Between FEC Block Loss of 1 hour

Operation Points	Ideal (see note)	MBMS FEC
Low Bitrate (CS3) (0.1 % BLER) 28.8 kbit/s	5 s: 24.9 kbit/s 20 s: 26.6 kbit/s	5 s: 24.4 kbit/s 20 s:26.2 kbit/s
Medium Bitrate (MCS-6) (0.5 % BLER) 59.2 kbit/s	5 s: 51.1kbit/s 20 s: 54.6 kbit/s	5 s: 50.4 kbit/s 20 s: 54.2 kbit/s
High Bitrate (MCS-9) (1 % BLER) 118.4 kbit/s	5 s: 99.5 kbit/s 20 s: 104.5 kbit/s	5 s: 98.2 kbit/s 20 s: 104.2 kbit/s
High Bitrate and High Error Rate (MCS-6) 10 % BLER, 146 byte packet payloads 118.4 kbit/s	5 s: 66.5 kbit/s 20 s: 72.5kbit/s	5 s: 66.4 kbit/s 20 s: 72.3 kbit/s
NOTE: This column provides the best possible performance which could be achieved with a theoretical "ideal" forward error correction code.		

Annex B: Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2006-03	31				Presented at SA#31 Plenary for approval	2.0.0	6.0.0
2006-03					Correct 3GPP logo, make capitalization consistent	6.0.0	6.0.1
2006-09	33	SP-060593	0001	2	Correcting the Associated Delivery Procedure Example	6.0.1	6.1.0
2007-06	36				Version for Release 7	6.1.0	7.0.0
2008-12	42				Version for Release 8	7.0.0	8.0.0
2009-12	46				Version for Release 9	8.0.0	9.0.0
2011-03	51				Inclision of LTE logo	9.0.0	9.0.1

History

Document history		
V9.0.0	January 2010	Publication (Withdrawn)
V9.0.1	May 2011	Publication