File and Stream Distribution for Mobile Broadcast Services

Candidate Version 1.3 – 14 Jan 2014

Open Mobile Alliance

OMA-TS-BCAST_Distribution-V1_3-20140114-C

Contents

Figures

Tables

1. Scope

The scope of this specification is the stream and file distribution functionality of the OMA Mobile Broadcast (BCAST) Enabler. Referring to the OMA BCAST functional architecture [BCAST11-Architecture], this document normatively specifies the interfaces FD-1, FD-2, FD-5, FD-6, SD-1, SD-2, SD-5 and SD-6.

Media codecs and related media-specific transport payloads are outside the scope of this specification.

2. References

2.1 Normative References

[3GPP TS 26.234] "Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs", 3rd Generation Partnership Project, Technical Specification 3GPP TS 26.234 Release 8,
URL: http://www.3gpp.org/

[3GPP TS 26.346] "Multimedia Broadcast/Multicast Service (MBMS); Protocols and Codecs", 3rd Generation Partnership Project, Technical Specification 3GPP TS 26.346 Release 8,
URL: http://www.3gpp.org/

[3GPP2 C.S0046-0] "3G Multimedia Streaming Services". 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0046-0, Release 0, Version 1.0, February 2006,
URL : http://www.3gpp2.org/Public_html/specs/tsgc.cfm

[BCAST10-XMLSchema-FD-AssociatedProcedure] "Mobile Broadcast Services – XML Schema for Associated Delivery Procedures – File Distribution", Open Mobile Alliance™, OMA-SUP-XSD_bcast_fd_associatedprocedure-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-FD-Backend] "Mobile Broadcast Services – XML Schema for File Distribution Backend Interfaces", Open Mobile Alliance™, OMA-SUP-XSD_bcast_fd_backend-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-FD-FDT] "Mobile Broadcast Services – XML Schema for File Description Table", Open Mobile Alliance™, OMA-SUP-XSD_bcast_fd_fdt-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-FD-ReceptionReport] "Mobile Broadcast Services – XML Schema for Reception Reporting – File Distribution", Open Mobile Alliance™, OMA-SUP-XSD_bcast_fd_receptionreport-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-SD-AssociatedProcedure] "Mobile Broadcast Services – XML Schema for Associated Delivery Procedures – Stream Distribution", Open Mobile Alliance™, OMA-SUP-XSD_bcast_sd_associatedprocedure-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-SD-Backend] "Mobile Broadcast Services – XML Schema for Stream Distribution Backend Interfaces", Open Mobile Alliance™, OMA-SUP-XSD_bcast_sd_backend-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-XMLSchema-SD-ReceptionReport] "Mobile Broadcast Services – XML Schema for Reception Reporting – Stream Distribution", Open Mobile Alliance™, OMA-SUP-XSD_bcast_sd_receptionreport-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST13-BCMCS-Adaptation] "BCAST Distribution System Adaptation – 3GPP2/BCMCS", Open Mobile Alliance™, OMA-TS-BCAST_BCMCS_Adaptation-V1_3,
URL: http://www.openmobilealliance.org/

[BCAST13-DVBH-IPDC-Adaptation] "BCAST Distribution System Adaptation – IPDC over DVB-H", Open Mobile Alliance™, OMA-TS-BCAST_DVB_Adaptation-V1_3,
URL: http://www.openmobilealliance.org/

[BCAST13-DVBNGH-Adaptation] "BCAST Distribution System Adaptation – DVB-Next Generation Handheld (NGH)", Open Mobile Alliance™, OMA-TS-BCAST_DVB_Adaptation-V1_3,
URL: http://www.openmobilealliance.org/

[BCAST13-DVBSH-IPDC-Adaptation] "BCAST Distribution System Adaptation – IPDC over DVB-SH", Open Mobile Alliance™, OMA-TS-BCAST_DVBSH_Adaptation-V1_3,
URL: http://www.openmobilealliance.org/

[BCAST13-FLO-Adaptation] "Broadcast Distribution System Adaptation – Forward Link Only", Open Mobile Alliance™, OMA-TS-BCAST_DVBSH_Adaptation-V1_3,
URL: http://www.openmobilealliance.org/

[BCAST13-MBMS-Adaptation] "BCAST Distribution System Adaptation – 3GPP/MBMS", Open Mobile Alliance™, OMA-TS-BCAST_MBMS_Adaptation-V1_3,
URL: http://www.openmobilealliance.org/

[BCAST13-Services] "Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_Services-V1_3, URL: http://www.openmobilealliance.org/

[BCAST13-SG] "Service Guide for Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_ServiceGuide-V1_3, URL: http://www.openmobilealliance.org/

[BCAST13-WiMAX-Adaptation] "BCAST Distribution System Adaptation – WiMAX", Open Mobile Alliance™, OMA-TS-BCAST_WiMAX_Adaptation-V1_3, URL: http://www.openmobilealliance.org/

[ETSI 102 472] ETSI TS 102 472 v1.1.1 (2006-04), "Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols", URL: http://portal.etsi.org/

[IOPPROC] "OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL: http://www.openmobilealliance.org/

[OMA Push] OMA-Push-V2_1 enabler. Open Mobile Alliance™. URL: http://www.openmobilealliance.org/

[RFC 2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, IETF RFC 2119, March 1997, URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC 2234] "Augmented BNF for Syntax Specifications: ABNF", D. Crocker, Ed., P. Overell, IETF RFC 2234, November 1997, URL: http://www.ietf.org/rfc/rfc2234.txt

[RFC 2246] "The TLS Protocol, Version 1.0", T. Dierks, C.Allen, IETF RFC 2246, January 1999, URL: http://www.ietf.org/rfc/rfc2246.txt

[RFC 2326] "Real Time Streaming Protocol (RTSP)", H. Schulzrinne, A. Rao, R. Lanphier, IETF RFC 2326, April 1998, URL : http://www.ietf.org/rfc/rfc2326.txt

[RFC 2616] "Hypertext Transfer Protocol -- HTTP/1.1", R. Fielding et. al, June 1999, IETF RFC 2616, URL: http://www.ietf.org/rfc/rfc2616.txt

[RFC 3450] "Asynchronous Layered Coding (ALC) Protocol Instantiation", M. Luby et. al, IETF RFC 3450, December 2002, URL: http://www.ietf.org/rfc/rfc3450.txt

[RFC 3451] "Layered Coding Transport (LCT) Building Block", M. Luby et. al, IETF RFC 3451, December 2002, URL: http://www.ietf.org/rfc/rfc3451.txt

[RFC 3452] "Forward Error Correction (FEC) Building Block", M. Luby et. al, IETF RFC 3452, December 2002, URL: http://www.ietf.org/rfc/rfc3452.txt

[RFC 3550] "RTP: A Transport Protocol for Real-Time Applications", H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, IETF RFC 3550, July 2003, URL: http://www.ietf.org/rfc/rfc3550.txt

[RFC 3695] "Compact Forward Error Correction (FEC) Schemes", M. Luby,. and L. Vicisano, IETF RFC 3695, February 2004, URL: http://www.ietf.org/rfc/rfc3695.txt

[RFC 3926] "FLUTE - File Delivery over Unidirectional Transport", T. Paila et. al, IETF RFC 3926, October 2004, URL: http://www.ietf.org/rfc/rfc3926.txt

[RFC 3986] "Uniform Resource Identifier (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, IETF RFC 3986, January 2005, URL: http://www.ietf.org/rfc/rfc3986.txt

[RFC4234] "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. October 2005, URL: http://www.ietf.org/rfc/rfc4234.txt

[SCRRULES] "SCR Rules and Procedures", Open Mobile Alliance™, OMA-ORG-SCR_Rules_and_Procedures, URL: http://www.openmobilealliance.org/

[SSL30] "SSL 3.0 Specification", Netscape Communications, November 1996,

URL: http://wp.netscape.com/eng/ssl3/draft302.txt

2.2 Informative References

[BCAST11-Architecture] "Mobile Broadcast Services Architecture", Open Mobile Alliance™, OMA-AD- BCAST-V1_1, URL: http://www.openmobilealliance.org/

[OMADICT] "Dictionary for OMA Specifications", Version x.y, Open Mobile Alliance™, OMA-ORG-Dictionary-Vx_y, URL:http://www.openmobilealliance.org/

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC 2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Definitions

ALC	The Asynchronous Layered Coding protocol for multiple rate congestion controlled reliable content delivery. The protocol is specifically designed to provide massive scalability using IP multicast as the underlying network service. [RFC 3450]
Associated Procedure Description	The configuration information for the associated delivery procedures
BCAST Distribution System	A system typically but not necessarily containing the ability to transmit the same IP flow to multiple Terminal devices simultaneously. A BCAST Distribution System (BDS) typically uses techniques that achieve efficient use of radio resources. A BDS consists of Network functionality up to the IP layer and optional Service Distribution/Adaptation functionality above the IP layer. Most BDSs support broadcast/multicast distribution in the network. Some BCAST Distribution Systems have the capability to deliver the IP flows in the network via unicast.
BCAST Service Distribution/Adaptation	BSD/A is responsible for the aggregation and delivery of BCAST services, and performs the adaptation of the BCAST Enabler to underlying BCAST Distribution System. [BCAST 11-Architecture]
BDS Service Distribution	BDSSD is an entity of the underlying BDS and it is responsible for the coordination and delivery of broadcast services to the BDS for delivery to the terminal, including file and stream distribution, and Service Guide distribution. [BCAST 11-Architecture]
Broadcast Service	A Broadcast Service is a “content package” suitable for simultaneous distribution to many recipients (potentially) without knowing the recipient. Either each receiver has similar receiving devices or the content package includes information, which allows the client to process the content according to his current conditions. [BCAST 11-Architecture]
Cachecast	A non-real-time file distribution service, for which the content could consist of audio, audio and video, and/or other types of data. Once the subscriber has subscribed to this service, the content is delivered to the subscriber’s mobile device in the background, during the Distribution Window, transparently to the user. The media is stored on the device and may be accessed by the user during a scheduled availability period referred to as the Presentation Window.
ClientID	The unique identifier for the FD-C. [BCAST 11-Architecture]
Content Creation	Source of content, may provide support for delivery paradigms (e.g. streaming servers); provides base material for content descriptions. [BCAST 11-Architecture]
ESI	Encoding Symbol ID identifies which specific encoding symbol(s) generated from the source block are carried in the packet payload.
FA	File Application Component [BCAST 11-Architecture]
FD	File Distribution Component [BCAST 11-Architecture]
FD-C	File Delivery - Client Component [BCAST 11-Architecture]
FEC	Forward error correction is a method of obtaining error control in data transmission in which the source (transmitter) sends redundant data and the destination (receiver) recognizes only the portion of the data that contains no apparent errors.
fileURI	Uniform resource identifier of the file

FLUTE	File Delivery over Unidirectional Transport is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. [RFC 3926]
GZIP	GNU compression utility. GZIP reduces the size of the named files using Lempel-Ziv LZ77 compression. Whenever possible, each file is replaced by one with the filename extension ".gz". Compressed files can be restored to their original form using gzip -d or gunzip or zcat.
LCT	Layered Coding Transport provides transport level support for reliable content delivery and stream delivery protocols. LCT is specifically designed to support protocols using IP multicast, but also provides support to protocols that use unicast. [RFC 3451]
NPT	NPT, or normal play time, is a timestamp indicating the stream absolute position relative to the beginning of the presentation.
Raptor	Raptor is a type of systematic forward error correction scheme known as a fountain code, in which as many encoding symbols as needed can be generated by the encoder from the source symbols. The decoder is able to recover the source symbols from any set of encoding symbols only slightly more in number than the number of source symbols.
RTCP	The RTP Control Protocol is the control protocol that works in conjunction with RTP. RTCP control packets are periodically transmitted by each participant in an RTP session to all other participants. Feedback of information to the application can be used to control performance and for diagnostic purposes. [RFC 3550]
RTP	The Real-time Transport Protocol defines a standardized packet format for delivering audio and video over the Internet. [RFC 3550]
SA	Stream Application Component [BCAST 11-Architecture]
SBN	Source Block Number identifies from which source block of the object the encoding symbols in the payload of the packet are generated.
SD	Stream Delivery Component [BCAST 11-Architecture]
SDP	Session Description Protocol is a format for describing streaming media session initialization parameters, and can also be used to describe file distribution sessions. [RFC 4566]
Service Guide	The information describing the Broadcast Services available to the End-User and the Terminal.
ServiceID	The globally unique identifier identifying the service.
ServiceURI	Specifies the service URI
Session Description	The Session Description is a Service Guide fragment which provides the session information for access to a service or content item. Further, the session description may contain auxiliary information.

3.3 Abbreviations

A/V	Audio/Visual
ABNF	Augmented Backus-Naur Form
ALC	Asynchronous Layered Coding
BDS	BCAST t Distribution System
BDS-SD/A	BDS Service Distribution/Adaptation
BSD/A	BCAST Service Distribution and Adaptation
CC	Content Creation
CENC	Common ENCryption
DASH	Dynamic Adaptive Streaming over HTTP
DVB-NGH	Digital Video Broadcast – Next Generation Handheld
DVB-T2	Digital Video Broadcast – Second Generation Terrestrial
eMBMS	Evolved Multimedia Broadcast Multicast Service
ESI	Encoding Symbol Identifier

EXT_CENC	FDT Instance Content Encoding Header
EXT_FDT	FDT Instance Header
EXT_FTI	FEC Object Transmission Information extension
FA	File Application Component
FD	File Delivery Component
FD-C	File Delivery - Client Component
FDT	File Delivery Table
FEC	Forward Error Correction
FEC-OTI	FEC-Object Transmission Information
FLUTE	File Delivery over Unidirectional Transport
FTP	File Transfer Protocol
HDR_LEN	(LCT) header length
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LCT	Layered Coding Transport
LTE	Long Term Evolution
MPEG	Motion Pictures Expert Group
NPT	Normal Play Time
NTP	Network Time Protocol
PSS	Packet-switched Streaming Services
Rack	Reception Acknowledgment
RFC	Request For Comments
RO	Rights Object
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
SA	Stream Application Component
SBN	Source Block Number
SD	Stream Delivery Component
SD-C	Stream Delivery-Client Component
SDP	Session Description Protocol
SGDD	Service Guide Delivery Descriptor
SGDU	Service Guide Delivery Unit
STaR	Statistical Reception for successful reception
TCP	Transmission Control Protocol
TOI	Transmission Object Identifier
TSI	Transmission Session Identifier
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

XML	Extensible Markup Language

4. Introduction

This specification defines the stream and file distribution functionality for OMA Mobile Broadcast Services. Stream distribution provides the delivery of real-time audio/visual streams from a network server to terminals. File distribution has similar functionality to stream distribution with the difference that instead of real-time media streams discrete media objects and files are delivered within the session. In the context of Mobile Broadcast Service, the assumption is that a single stream or file delivery session distributes the content to multiple recipients simultaneously. The use of the Interaction Channel to realize the associated procedures for stream and file distribution is also specified in this document.

4.1 Version 1.0

This specification is agnostic to the underlying IP-based BCAST Distribution System. It specifies common ways to achieve the delivery of files and streams over various such IP-based broadcast distribution technologies. Therefore, it provides a greater level of interoperability.

4.2 Version 1.1

In version 1.1, BCAST added the technical solution for hybrid scenario. Hybrid scenario is to use broadcast channel as well as interactive channel for service and content delivery. The related sections of main body of TS-Distribution were updated and to provide more precious information, new appendix D for hybrid scenario was added. In addition, version of some normative references was updated according to the provision of latest versions.

4.3 Version 1.2

The Mobile Broadcast Services Enabler 1.2 focuses on defining the adaptation to the DVB-NGH standard. It could be extended to other second generation DVB broadcast bearers, such as DVB-T2 and mobile and terrestrial profile T2-Lite, when the DVB Generic Stream Encapsulation Logical Link Control (GSE LLC) is used for the IP encapsulation. It aims to adopt the BCAST 1.2 specifications as widely as possible.

4.4 Version 1.3

BCAST 1.3 aims to

- Define the delivery of MPEG DASH-based contents in BCAST systems and the appropriate signalling for the use of BCAST service and content protection systems to protect these MPEG DASH-based contents.
- Allow the Common Encryption (CENC) defined by MPEG as possible encryption method to be used for BCAST
- Extend the use of MBMS BDS to the last releases of 3GPP MBMS specifications for the use of BCAST over e-MBMS for LTE networks.

5. File Distribution

5.1 Introduction

The specification for the OMA BCAST File Distribution function consists of the specification of four interfaces: FD-1, FD-2, FD-5 and FD-6. The interfaces FD-5 and FD-6 are terminal-network interfaces and the functional entities across these interfaces are the File Delivery Client Component (FD-C) on the terminal side and File Delivery Component (FD) on the network side. These interfaces are specified in sections 5.2 and 5.3. The interfaces FD-1 and FD-2 are back-end interfaces within the system(s) serving the OMA Mobile Broadcast Services and the functional entities across these interfaces are the File Distribution Component (FD) and the File Application (FA), both on the network side. These back-end interfaces are specified in section 5.4. Lastly, file distribution over the Interaction Channel is specified in section 5.5 and file distribution in a hybrid environment is specified in section 5.6.

5.2 File Distribution over Terminal-Network Interfaces

The OMA BCAST File Distribution function uses the ALC protocol [RFC 3450] when delivering content in files over interface FD-5. Use of ALC protocol as described in this clause is REQUIRED for both FD-C and FD.

ALC combines the Layered Coding Transport (LCT) building block [RFC 3451], a congestion control building block, and the Forward Error Correction (FEC) building block [RFC 3452] 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], congestion control is not appropriate in the type of environment in which BCAST File Distribution is provided, and thus congestion control is not used for the BCAST File Distribution function. ALC is carried over UDP/IP, and is independent of the IP version and the underlying link layers used.

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 for the broadcast channel. The FEC building block allows the choice of an appropriate FEC code to be used within ALC, including using the “Compact No-Code” FEC scheme [RFC 3695] (FEC Encoding ID 0, also known as “Null-FEC”) that simply sends the original data using no FEC coding. In addition to FEC protection over the broadcast channel, the OMA BCAST File Distribution function specifies point-to-point post-delivery methods to provide additional robustness when full reliability for file delivery is required. The post-delivery methods are executed over interface FD-6.

The metadata associated with files (name, URL, media type, etc.) can be delivered in two ways:

- The FD-C SHALL support the delivery of metadata associated with files in-band within the file delivery session, in which case the Transport Object Identifier 0 carries the File Delivery Table, and the file delivery session is a FLUTE session as specified in [3GPP TS 26.346].
- The FD-C SHALL support the delivery of metadata associated with files in the Service Guide as specified in section 5.1.2.4 of the OMA BCAST Service Guide [BCAST13-SG].

The FD SHALL support either one or both of the above-mentioned methods for signalling the file parameters.

5.2.1 Content Encoding

Files MAY be content encoded using the generic GZIP algorithm [RFC 1952]. Terminals SHALL support GZIP content decoding of files.

For GZIP-encoded files, the “Content-Encoding” attribute of the file description (in the FDT or in the Service Guide) SHALL be given the value “gzip”.

5.2.2 Forward Error Correction Building Block

The "Compact No-Code” FEC scheme [RFC 3695] (FEC Encoding ID 0, also known as "Null-FEC") SHALL be supported.

In addition, the Raptor FEC scheme (FEC encoding ID 1) MAY be supported. The Raptor FEC scheme is specified in [3GPP TS 26.346] for MBMS, and in [ETSI TS 102 472] for DVB-H IPDC.

5.2.3 File Descriptions

The delivery of a file is declared by one or several file descriptions. Each file description establishes a mapping between the file URI, a URI uniquely identifying the file during file distribution, and a Transmission Object Identifier (TOI) identifying one transport object in the file distribution session. More specifically, file descriptions are defined as follows:

- For a file delivered in a FLUTE session: the file description is any FDT Instance delivered in this session that describes this file, i.e. that contains a 'File' element with 'Content-Location' attribute set to the file URI. For such file descriptions, the version is the wrap-around adjusted value of the FDT Instance ID (carried in the FDT Instance Header - EXT_FDT), and the expiry time is the 'Expires' attribute value of the FDT Instance.
- For a file delivered in an ALC-only session: the file description is a structure (delivered in this session or by other means) providing the TOI, among other information, of the transport object in the session (meaning that structures providing some information on the file, but not the TOI, cannot be used to determine end of file and end of session). For such file descriptions, the version is typically the wrap-around adjusted value of a "version" attribute, and the expiry time a "validTo" attribute value.

  Note: For ALC-only delivery sessions, possible file descriptions include 'Access' fragments of the service guide (that can declare the TOI of any type of file), and 'InteractivityMediaDocument' elements (that declare the TOI of 'MediaObjectSet' file bundles).

5.2.4 File Versioning

A file (URI) MAY be associated with several transport objects (i.e. with several TOI values) during the lifetime of the file distribution session. In this case, the transport object declared by the file description with the highest version number SHALL represent the latest version of the file. A new file description MAY keep the TOI associated with a given file unchanged, which means that the version of the file did not change.

Note: The FD-C SHOULD not send post-repair requests for an old version of a file once a file description declaring a newer version of the file is received.

5.2.5 Signalling End of File and End of Session

5.2.5.1 Signalling End of File Delivery Session

A file delivery session is considered complete when one of the following events occurs:

- The delivery session is declared by an SDP-formatted session description, the stop time provided for this session is bounded (i.e. second sub-field of "t=" field is not null), and this stop time is reached.
- The delivery session is described by a Notification message via a 'SessionInformation' element, this element contains a 'validTo' attribute, and the end time expressed in 'validTo' is reached.
- The FD-C receives an end-of-session packet (ALC packet with A-flag in LCT header set to true).
- The terminal decides to exit the session.

5.2.5.2 Signalling End of File Delivery

The FD-C SHALL determine that the delivery of a file has ended when one of the following events occur:

- The FD-C determines that the file delivery session has ended, as specified in section 5.2.5.1.
- The FD-C receives an end-of-object packet (ALC packet with B-flag in LCT header set to true) for the transport object representing the latest version of the file. The FD-C SHALL NOT determine that the delivery has ended for this file in case the end-of-object packet belongs to an older version of the file.
- For cachecast files, when the time represented by the sum of 'endTime' and 'duration' of 'DistributionWindow', as specified in the 'Schedule' fragment, has elapsed.

The FD-C SHOULD also determine that the delivery of a file has ended when:

- In the set of file descriptions describing the latest version of the file, the last file description to expire has expired.

  The expiry time of this file description, when handled as the anticipated end time of delivery for this file, is therefore updated when the FD-C receives a file description describing a newer version of the file, or a file description describing the latest version of the file and expiring later than the formerly computed expiry time.

This end of file delivery (as concluded by the FD-C) is the starting point of any associated delivery procedure requested or needed for this file. When applicable, these procedures SHOULD be launched for the latest version of the file only, and not for older versions of the file.

5.2.6 Signalling of Parameters with FLUTE

5.2.6.1 Signalling of Parameters with Basic ALC/FLUTE Headers

LCT mandatory header fields for FLUTE and ALC SHALL be as specified in [RFC 3926], [RFC 3450] 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), or 32 bits (S=1, H=0) when TOI is an identifier of 32 bits.
- The Transmission Object Identifier (TOI) field SHOULD be of length 16 bits (O=0, H=1) or 32 bits (O=1, H=0). The maximum length of Transport Object Identifier (TOI) field SHOULD be 32 bits.
- The terminal SHALL support a TOI field length up to 112 bits.
- Only TOI 0 (zero) SHALL be used for FDT Instances.
- The following features MAY be used for signalling the end of session and the 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:

- If the LCT header provides the Sender Current Time present flag (T), it SHALL be set to zero.
- If the LCT header provides the Expected Residual Time present flag (R), it 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 the “Compact No-Code” FEC scheme, the payload ID SHALL be set according to [RFC 3695] such that a 16-bit SBN (Source Block Number) and then the 16-bit ESI (Encoding Symbol ID) are given.
- For the Raptor FEC scheme, the payload ID SHALL consist of a 16-bit source block number (SBN) and a 16-bit encoding symbol ID (ESI).

5.2.6.2 Signalling of Parameters with LCT Extension Header

In order to provide timing information related to an ALC/FLUTE session:

- the network MAY use the EXT_TIME LCT extension header, and
- the terminal MAY support the EXT_TIME LCT extension header.

5.2.6.3 Signalling of Parameters with FLUTE Extension Headers

FLUTE extension header fields EXT_FDT, EXT_FTI, EXT_CENC [RFC 3926] SHALL be used as follows:

- EXT_FTI SHALL be included in every FLUTE packet carrying symbols belonging to any FDT Instance.
- 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.

The optional use of EXT_FTI for packets carrying symbols of files (not FDT instances) SHALL comply to FLUTE [RFC 3926] for the signalling of FEC Object Transmission Information associated to FEC Encoding 0.

5.2.6.4 Signalling of Parameters with FDT Instances

The FLUTE FDT Instance schema defined in section 5.2.6.6 SHALL be used. In addition, the following applies to both the FDT-Instance 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).

The inclusion of the following FDT Instance data elements is optional and depends on the FEC Scheme:

- FEC-OTI-Maximum-Source-Block-Length;
- FEC-OTI-Encoding-Symbol-Length;
- FEC-OTI-Max-Number-of-Encoding-Symbols;
- FEC-OTI-Scheme-Specific-Info

These optional FDT Instance data elements MAY be included for FLUTE in BCAST:

- Complete (the signalling that an FDT Instance provides a complete, and subsequently unmodifiable, set of file parameters for a FLUTE session MAY be performed according to this method);
- Content-Length (source file length in bytes);
- Content-Type (content MIME type);
- FEC-OTI-FEC-Encoding-ID;
- FEC-OTI-FEC-Instance-ID;
- Content_Encoding;
- Transfer_length;
- Content-MD5 (It is recommended to indicate the MD5 hash value whenever multiple versions of the file, i.e. distinct file objects identified by the same Content-Location, are anticipated for the download session(s). Note that in case a file object is content encoded using gzip and also has an associated MD5 hash value provided in Content-MD5, the MD5 hash value is calculated over the file as transported by FLUTE, i.e. after any gzip compression as indicated by

Content-Encoding has been applied. For the terminal, this implies that the MD5 hash value under these circumstances is calculated before gzip decompression.)

These optional BCAST FDT Instance extension elements MAY be included for FLUTE in BCAST:

- Version-ID-Length;
- MBMS-Session-Identity;
- MBMS-Session-Identity-Expiry.

5.2.6.5 Extensions to FLUTE FDT Instances

BCAST extends the IETF FLUTE FDT Instances with the following elements and attributes:

The 'Version-ID-Length' attribute MAY be included in the 'FDT-Instance' and/or in the 'File' element to signal the use of split-TOI mechanism for the TOI(s) in scope of the element including the attribute. The attribute value specifies the number of least significant bits allocated to the Version_ID part in the TOI(s), with the remaining bits (the most significant bits) consisting of the Object_ID part.

The 'MBMS-Session-Identity' element MAY be included in the 'File' element to associate 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 MAY be included in the 'FDT-Instance' element to associate 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) time value. These 32 bits provide an unsigned integer representing the time in seconds relative to 0 hours 1 January 1900.

5.2.6.6 FLUTE FDT Instance XML Schema

5.2.6.6.1 XML Schema (Normative)

The syntax of the BCAST FLUTE FDT Instance is specified by the XML schema provided in [BCAST10-XMLSchema-FD-FDT].

In this XML schema, multiple types are specified for the 'FDT-Instance' and/or the 'File' element catering for the specifics of the different BDSs. The table below specifies which type is to be assigned to these elements via the 'xsi:type' attribute in case it is wanted to apply some or all BDS attribute restrictions to a given BCAST FDT Instance:

Applied restrictions	Type of <FDT-Instance>	Type of <File >
MBMS restrictions, or MBMS + DVB-H restrictions	"FDT-InstanceType-BdsMbmsDvb"	"FileType-BdsMbmsDvb"
DVB-H restrictions	"FDT-InstanceType-BdsDvb"	
No BDS restrictions check	"FDT-InstanceType" (typically omitted since default)	"FileType" (typically omitted since default)

Table 1: 'xsi:type' attribute value in BCAST FDT Instances

5.2.6.6.2 XML Instance Example (Informative)

The following FDT Instance example illustrates these concepts: mixed BDS extensions in the same element, inclusion of other elements/attributes from other namespaces, and attribute presence control (via the 'xsi:type' attribute) applying the underlying BDS's constraints.

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

```
    xmlns="urn:oma:xml:bcast:fd:fdt:1.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:foo="urn:foo"
    xsi:schemaLocation="http://www.openmobilealliance.org/tech/profiles/bcast_fd_fdt-v1_0.xsd"
    Complete="true"
    Content-Encoding="gzip"
    Content-Type="video/3gpp"
    FEC-OTI-Encoding-Symbol-Length="512"
    Expires="331129600"
    FEC-OTI-FEC-Encoding-ID="0"
    xsi:type="FDT-InstanceType-BdsMbmsDvb">
    <File
        xsi:type="FileType-BdsMbmsDvb"
        Content-Type="application/sdp"
        Content-Length="7543"
        Transfer-Length="4294"
        Version-ID-Length="8"
        TOI="2"
        FEC-OTI-Encoding-Symbol-Length="16"
        Content-Location="http://www.example.com/fancy-session/main.sdp">
        <MBMS-Session-Identity>93</MBMS-Session-Identity>
        <foo:yousay>goodbye</foo:yousay>
    </File>
    <File
        xsi:type="FileType-BdsMbmsDvb"
        Content-Length="161934"
        Transfer-Length="157821"
        TOI="3"
        FEC-OTI-FEC-Encoding-ID="1"
        FEC-OTI-Encoding-Symbol-Length="512"
        FEC-OTI-Scheme-Specific-Info="AAEBBA=="
        Content-Location="http://www.example.com/fancy-session/trailer.3gp"
        foo:myattribute="myvalue">
        <MBMS-Session-Identity>93</MBMS-Session-Identity>
    </File>
    <MBMS-Session-Identity-Expiry value="3311288760">93</MBMS-Session-Identity-Expiry>
    <foo:andisay>hello</foo:andisay>
</FDT-Instance>
```

5.3 Associated Procedures for File Distribution

Associated delivery procedures describe general procedures, which start before, during or after the BCAST data transmission phase. They provide auxiliary features to BCAST services in addition to, and in association with, BCAST content delivery and content delivery sessions. Those procedures that SHOULD only be permitted after the BCAST data transmission phase MAY also be described as post-delivery procedures.

The present document describes two associated delivery procedures:

- File repair, for post-delivery repair of files initially delivered as part of a BCAST download session.
- Content reception reporting of files delivered to a BCAST terminal.

These procedures are enabled by establishing a point-to-point connection to communicate the context (e.g. file and session in question) to the network and the BCAST service 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 BCAST terminals (File Delivery Clients) SHALL be distributed over time and resources (network elements).

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

FD and FD-C SHOULD support the download reception reporting procedure as defined in section 5.3.2.

FD and FD-C SHOULD support the download file repair procedure as defined in section 5.3.3.

5.3.1 Associated Procedure Description

An associated procedure description instance (configuration information) describing associated delivery procedures MAY be delivered to the BCAST terminals:

- In the Service Guide prior to the BCAST download delivery session along with the session description (out-of-band of that session); or
- In-band within a BCAST download delivery session.

The most recently delivered configuration file (i.e. the one with the highest version number) 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, override 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.

In the Service Guide, the associated procedure description instance is clearly identified using a URI, to enable the FD-C to cross-reference in-band and out-of-band configuration files.

The MIME application type “application/vnd.oma.bcast.associated-procedure-parameter+xml” 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 [BCAST10-XMLSchema-FD-AssociatedProcedure].

5.3.2 Reception Reporting

Following successful reception of content, a reception reporting procedure MAY be initiated by the FD-C to the FD.

For BCAST download delivery, the reception reporting procedure is used to report the complete reception of one or more files.

If the FD provided parameters requiring reception reporting confirmation then the FD-C SHALL confirm the content reception.

If reception reporting is requested for statistical purposes the FD MAY specify the percentage subset of FD-Cs it would like to perform reception reporting.

Transport errors can prevent the FD-C 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. The AFD-C SHALL behave according to the information it has even when it is aware that this MAY be incomplete.

The FD-C:

1. Identifies the complete reception of a content item (e.g. a file). See sections 5.3.2.1and 5.3.2.2.
2. Determines the need to report reception. See section 5.3.2.3.
3. Selects a time (request time) at which a reception report request will be sent and selects an FD from a list - both randomly and uniformly distributed. See sections 5.3.2.4 and 5.3.2.5.
4. Sends a reception report request message to the selected FD at the selected time. See section 5.3.2.6.

Then the FD:

1. Responds with an HTTP response message either describing the success of the reporting operation, or alternatively, describing an error case. See section 5.3.2.7.

5.3.2.1 Identifying Complete File Reception from BCAST Download

The FD-C SHALL determine that a file has been completely downloaded when it is fully received and reconstructed by BCAST reception and/or a subsequent File Repair Procedure (section 5.3.3). The purpose of determining file download completeness is to determine when it is feasible for a FD-C to compile the RAck reception report for that file.

5.3.2.2 Identifying Complete BCAST Delivery Session Reception

Delivery sessions (download and streaming) are considered complete when the "stop time" value of the session description (from "t=" in SDP) is reached. Where the end time is unbounded (<stop time> = 0) then this parameter is not used for identifying completed sessions.

Delivery sessions are also considered complete when the terminal decides to exit the session - where no further data from that session will be received.

5.3.2.3 Determining Whether a Reception Report Is Required

Upon full reception of a content item or when a session is complete, the FD-C 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.

Each BCAST file delivery or stream delivery session MAY be associated with zero or one associated delivery procedure descriptions. Where an associated delivery procedure description is associated with a session, and the description includes a ‘postReceptionReport’ element, the FD-C 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.

One of the following will determine the FD-C 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.

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 FD-Cs 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 and StaR-all, but SHALL NOT be used with RAck.

When the ‘samplePercentage’ is not present or its value is 100 each FD-C which entered the associated session SHALL send a reception report. If the ‘samplePercentage’ is provided for reportType StaR and StaR-all and the value is less than 100, the FD-C generates a random number which is uniformly distributed in the range of 0 to100. The FD-C sends the reception report when the generated random number is of a lower value than the ‘samplePercentage’ value.

5.3.2.4 Request Time Selection

The FD-C selects a time at which it is to issue a delivery confirmation request.

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 section 5.3.3.3. 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. The default behaviour is that a FD-C SHALL stop its post-reception report timers which are active when a file repair timer expires, which results in the successful initiation of point-to-point communications between FD-C and FD.

In some circumstances, the system bottleneck MAY be in the FD 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 FD-C 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.

For StaR and StaR-all, session completeness - according to section 5.3.2.1 - SHALL determine the back-off timer initialization time.

For RAck, the complete download session - according to section 5.3.2.1 - 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 according to section 9.4.1.

5.3.2.5 Reception Report Server Selection

Reception report server selection is performed according to the procedure described in section 5.3.3.4.

5.3.2.6 Reception Report Message

Once the need for reception reporting has been established, the FD-C sends one or more Reception Report messages to the FD. All Reception Report requests and responses for a particular BCAST transmission SHOULD take place in a single TCP session using the HTTP protocol [RFC 2616].

The Reception Report request SHALL include the URI of the file for which delivery is being confirmed. The URI is required to uniquely identify the file (resource).

The FD-C SHALL make a Reception Report request using an HTTP POST request carrying XML formatted metadata for each reported received content (file). If more than one file was downloaded in a particular BCAST file delivery session, multiple descriptions SHALL be added in a single POST request.

Each Reception Report is formatted in XML according the XML schema in [BCAST10-XMLSchema-FD-ReceptionReport].

For Reception Acknowledgement (RAck) a ‘receptionAcknowledgement’ element SHALL provide the relevant data.

For Statistical Reporting (StaR) a ‘statisticalReporting’ element SHALL provide the relevant data.

For both RAck and StaR/StaR-all (mandatory):

- For download, one or more ‘fileURI’ elements SHALL specify the list of files which are reported.

For only StaR/StaR-all (all optional unless otherwise stated):

- 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.
- The ‘sessionID’ attribute identified the delivery session. This is of the format source_IP_address + ":" + TSI/RTP_source_port.
- The ‘sessionType’ attribute defines the basic delivery method session type used. The format is "download" || "streaming" || "mixed".
- The ‘serviceId’ attribute value and format is taken from the respective userServiceDescription serviceID definition.
- The ‘clientId’ attribute is unique identifier for the FD-C.

- The ‘serviceURI’ attribute value and format is taken from the respective ‘associatedDeliveryProcedureDescription’ element’s ‘serviceURI’ child element which was selected by the FD-C for the current report. This attribute expresses the reception report server to which the reception report is addressed.

5.3.2.7 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].

5.3.3 File Repair

The text in this section and its sub-sections describes file repair for the case where information about downloaded files is carried in-band in a FLUTE download session. The description needs to be complemented with the case when file information is delivered in the service guide rather than in the FLUTE FDT table.

5.3.3.1 Introduction

The purpose of the File Repair Procedure is to repair lost or corrupted file fragments from BCAST file delivery sessions. When in a multicast/broadcast environment, scalability becomes an important issue as the number of BCAST terminals with support for the FD-C grows. Three problems must generally be avoided:

- Feedback implosion due to a large number of FD-Cs 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 FD-C requests.
- File repair server overload, caused again by the incoming and outgoing traffic due to the FD-Cs’ requests arriving at the FD, and the FD 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 BCAST file repair.

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

The FD-C:

1. Identifies the end of transmission of files or sessions.
2. Identifies the missing data from a BCAST 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 BCAST download session of repair data is configured in the associated delivery descriptions, a FD-C SHOULD wait for repair data in the defined BCAST download session.

Then the file repair server:

1. Responds with a repair response message either containing the requested data, redirecting the FD-C to an BCAST download session, redirecting the FD-C to another file repair server, or alternatively, describing an error case.

The FD MAY also send the repair data in the BCAST download session.

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.

5.3.3.2 Identification of Missing Data from a BCAST Download

The session description and the BCAST download delivery protocol provide the FD-C with sufficient information to determine the source block and encoding symbol structure of each file. From this a FD-C is able to determine which source symbols SHOULD have been transmitted but have not been received. The FD-C 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 FD-C is able to identify any source symbols lost in transmission, and the number of required source and/or repair symbols that would complete the reconstruction of a source block (of a file).

5.3.3.3 Back-off Timing Procedure

This section describes a back-off mode for BCAST download to provide information on when a FD-C, that did not correctly receive some data from the BCAST sender during a transmission session, can start a request for a repair session. In the following it is specified the method by which the FD-C calculates a time (back-off time), instance of the back-off mode, to send a file repair message to the FD.

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 FD-C.

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 sections.

A FD-C SHALL generate random or pseudo-random time dispersion of repair requests to be sent from the FD-C to the FD. 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 FD-C) enhances system scalability to the total number of such messages the system can handle without failure.

5.3.3.3.1 Offset Time

The offset time refers to the repair request suppression time to wait before requesting repair, or in other words, it is the time that a FD-C SHALL wait after the end of the BCAST 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.

5.3.3.3.2 Random Time Period

The random time period refers to the time window length over which a FD-C 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 FD-C SHALL calculate a uniformly distributed random time out of the interval between 0 and random time period.

5.3.3.3.3 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 BCAST data transmission. The FD-C SHALL NOT start sending the repair request message before this calculated time has elapsed after the initial transmission ends.

5.3.3.3.4 Reset of the Back-off Timer

The reception of an updated (higher version number) ‘associatedDeliveryProcedureDescription’ element 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.

5.3.3.4 File Repair Server Selection

5.3.3.4.1 List of Server URIs

A list of file repair service URIs is provided by a list of server URIs as elements of the associated delivery procedure description in the ‘postFileRepair’ element. The service URIs host identity MAY also be given as IP addresses. The file repair server 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 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 via the "xs:anyURI" value. At least one ‘serviceURI’ element SHALL be present.

5.3.3.4.2 Selection from the Server URI List

The FD-C SHALL randomly select one of the service URIs from the list, with uniform distribution.

5.3.3.5 File Repair Request Message

Once missing file data is identified, the FD-C sends one or more messages to a file repair server requesting transmission of data that allows recovery of the missing file data. All file repair requests and responses for a particular BCAST transmission SHALL take place in a single TCP session using the HTTP protocol [RFC 2616]. The repair request is routed to the file repair server IP address resolved from the selected ‘serviceURI’ element.

The timing of the opening of the TCP connection to the server, and the first repair request, of a particular FD-C is randomized over a time window as described in section 5.3.3.3.2. If there is more than one repair request to be made these are sent immediately after the first.

When a FD-C identifies symbols in repair requests these 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 FD MAY use these and/or redundant symbols in providing the necessary repair data.

5.3.3.5.1 File Repair Request Message Format

After the BCAST download session, the FD-C identifies a set of symbols, which allows recovery of the missing file data and requests for their transmission in a file repair session. Specific encoding symbols are uniquely identified by the combination (URI, SBN, ESI).

The file repair request SHALL include the URI of the file for which it is requesting the repair data. URI is required to uniquely identify the file (resource) and is found from either the FLUTE FDT or the Access fragment describing the file. Additionally, the repair request SHALL contain an indication of the MD5 hash value of the file whenever this value is provided to the terminal (which, besides message integrity check, is used to identify a specific version of the file).Content-MD5 is also found from either the FLUTE FDT or the Access fragment describing the file. The (SBN, ESI) pair uniquely identifies an encoding symbol. For completely missed files, a Repair Request MAY give only the URI of the file and optionally the base 64 encoded 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.

The FD-C makes a file repair request using the HTTP [RFC 2616] request method GET. If specific symbols are requested, the (SBN, ESI) of the requested encoding symbols are URL-encoded into the query part of the URI [RFC 3986] 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 that is subsequent in the symbol sequence to the latest received symbol for that source block and the number of symbols requested.

For example, assume that in a BCAST download session a file with URI = www.example.com/news/latest.txt was delivered to an FD-C. After the BCAST download session, the FD-C recognized that it did not receive two packets with SBN = 5, ESI = 12 and SBN=20, ESI = 27. If the repair service URI (from the associated delivery procedure description) is "http://bcastrepair1.example.com/path/repair_script" and the Base64-encoded MD5 value of that file is "ODZiYTU1OTFkZGY2NWY5OD==",then the HTTP GET request is as follows:

GET /path/repair_script?fileURI=www.example.com/news/latest.txt&Content-MD5=ODZiYTU1OTFkZGY2NWY5OD==&SBN=5;ESI=12&SBN=20;ESI=27 **HTTP/1.1**

Host: bcastrepair1.example.com

A file repair session SHALL be used to recover the missing file data from a single BCAST download session only. If more than one file was downloaded in a particular BCAST download session, and, if the FD-C needs repair data for more than one file received in that session, the FD-C SHALL send separate HTTP GET requests for each file.

An HTTP implementation within the FD-C 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 FD-C SHALL distribute the URL-encoded data into multiple HTTP GET requests.

In any case, all the HTTP GET requests 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, the details of the syntax used for the above request method are presented using ABNF.

In this case an HTTP GET with a normal query SHALL be used to request the missing data, according to HTTP/1.1 [RFC 2616].

The general HTTP URI syntax is as follows [RFC 2616]:

- repair_request_http_URL = repair_service_URI "?" query
- repair_service_URI = <selected serviceURI from the Associated Delivery Procedure Description>

Where, for BCAST File Repair Request:

- query = file_uri ["&" content_md5] *("&" sbn_info)
- file_uri = "fileURI=" URI-reference; URI-reference is as defined in [RFC 3986]
- 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

Thus, the following symbols adopt a special meaning for BCAST file repair 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://bcastrepair1.example.com/path/repair_script" is:

- http://bcastrepair1.example.com/path/repair_script?fileURI= www.example.com/greatmusic/number1.aac&Content-MD5=ODZiYTU1OTFkZGY2NWY5OD==&SBN=12;ESI=34

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:

- A symbol of a source block (like in the above example).
- 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 of source blocks (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).

5.3.3.6 File Repair Response Message

Once the BCAST file repair server has assembled a set of encoding symbols that contain sufficient data to allow the FD-C to reconstruct the file data from a particular file repair request, the BCAST file repair server sends one message to the FD-C. Each file repair response occurs in the same TCP and HTTP session as the repair request that initiated it.

An FD-C SHALL be prepared for any of these 4 response scenarios:

- The FD returns a repair response message where a set of encoding symbols forms an HTTP payload as specified below.
- The FD redirects the terminal to a broadcast/multicast delivery (an BCAST download session).
- The FD redirects the terminal to another file repair server (if a server is functioning correctly but is temporarily overloaded).
- An HTTP error code is returned (note that section 5.3.3.7 describes the case of no FD response).

For (reasonably) uniformly distributed random data losses, immediate point-to-point HTTP delivery of the repair data will generally be suitable for all FD-C. However, broadcast/multicast delivery of the requested data MAY be desirable in some cases:

- A repeat BCAST download (all or part of the files from a download session) is already scheduled and the FD prefers to handle repairs after that repeat BCAST download.
- Many FD-C 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 terminals that have made a repair request would be advantageous.

5.3.3.6.1 File Repair Response Messages Codes

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 section 5.3.3.7.2. If the FD-C 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 FD-C 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 BCAST (repair) session as described in section 5.3.3.7.3. Other HTTP status codes [RFC 2616] 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 not found), the server SHALL respond with "400 Bad Request" and optionally with "0001 File not found" in the response body. As a result, the FD-C MAY choose another file repair server as defined in clause 5.3.3.4.

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 FD-C MAY choose another file repair server as defined in clause 5.3.3.4. Or the FD-C MAY request the latest version of the file and discard the previously received chunks of the file. Note, the FD-C 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 response body. As a result, the FD-C 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 "BCAST1.0". As a result, the FD-C 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.

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 [RFC 2234] 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

5.3.3.6.2 File Repair Response Message Format for HTTP Delivery of Repair Data

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.
- Content type of the HTTP payload (see below).
- Content transfer encoding, set to binary.

The Content-Type header SHALL be set to “application/vnd.oma.bcast.simple-symbol-container”, which denotes that the message body is a simple container of encoding symbols as described below.

This HTTP message header is as follows:

- HTTP/1.1 200 OK
- Content-Type: application/vnd.oma.bcast.simple-symbol-container
- Content-Transfer-Encoding: binary

NOTE: Other HTTP headers [RFC 2616] 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 FD-C 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 1 illustrates the complete file repair response message format (box sizes are not indicative of the relative lengths of the labelled entities).

<table>
<tr><td colspan="3">HTTP Header</td></tr>
<tr><td>Length Indicator</td><td>FEC Payload ID</td><td>Encoding Symbols</td></tr>
<tr><td>Length Indicator</td><td>FEC Payload ID</td><td>Encoding Symbols</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>Length Indicator</td><td>FEC Payload ID</td><td>Encoding Symbols</td></tr>
</table>

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 1: File Repair Response Message Format

5.3.3.6.3 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 section 5.3.3.6.2.

The file repair server uses the HTTP response status code 302 (Found - Redirection) to indicate to the FD-C 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 use 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 ‘associatedDeliveryProcedureDescription’ element 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 FD-C 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’ element can not be assured in all cases.

The existence of a broadcast/multicast file repair session is signalled by the inclusion of the optional broadcast/multicast file repair 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 file repair session as the original download session, the FD-C simply SHALL NOT leave the group. Otherwise, the FD-C SHALL join to the BCAST bearer for the repair session as it would for any BCAST session.

A BCAST file repair session behaves just as a BCAST file delivery 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 BCAST file delivery method.

5.3.3.7 Server Not Responding Error Case

In the error case where the FD-C determines that the selected file repair server is not responding, it SHALL return to the server URI 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 FD-C SHALL then be immediately sent to the newly selected file repair server.

If all of the repair servers from the serviceURI list are determined to be not responding, the FD-C MAY attempt an HTTP GET to retrieve a, potentially new, instance of the session’s associated procedure description; otherwise if but the operation of HTTP post and HTTP GET failed.

The FD-C determines that a file repair server is not responding if any of these conditions apply:

- The FD-C is unable to establish a TCP connection to the server.
- The server does not respond to any of the HTTP repair requests that have been sent by the FD-C (it is possible that second and subsequent repair requests are sent before the first repair request is determined to be not-responded-to).
- The server returns an unrecognized message (not a recognizable HTTP response).
- The server returns an HTTP server error status code (in the range 500 to 505).

5.3.4 XML Schema for Associated Delivery Procedures

The formal XML syntax of associated delivery procedure description instances is specified in [BCAST10-XMLSchema-FD-AssociatedProcedure].

Note: This specification uses the element name ‘serviceURI’, however in the referenced schema and in conforming XML instances the element name ‘serverURI’ can alternatively be used.

The formal XML syntax of reception report request instances is specified in [BCAST10-XMLSchema-FD-ReceptionReport].

5.4 File Distribution over Back-end Interfaces

This section specifies interfaces between logical BCAST “back-end” entities. The specification is applicable if the interfaces are exposed in a BCAST implementation. If a BCAST implementation does not expose the interfaces, i.e, they are internal to that implementation, they MAY be realized using protocols and methods not specified here. All interfaces in this section are defined in the OMA BCAST Architecture document [BCAST11-Architecture].

5.4.1 Interface FD-1 and FD-2

Interface FD-1 between CC and FA provides the attributes of files as well as the files themselves for BCAST file distribution services.

Interface FD-2 between FA and FD provides the attributes of files as well as the files themselves for BCAST file distribution services.

5.4.1.1 Protocol Stacks

The protocol stack shown in Figure 2 SHALL be used for file delivery over back-end interfaces FD-1 between CC and FA and FD-2 between FA and FD. For secure operation over either of these backend interfaces, the entities implementing the interface SHALL support HTTPS, where HTTPS SHALL be based on SSL3.0 [SSL30] and TLS 1.0 [RFC 2246].

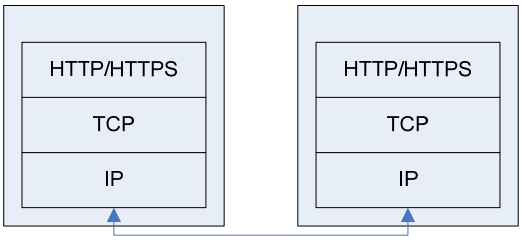

Figure 2: Protocol Stack for Back-end Interface of File Delivery

HTTP 1.1 over TCP/IP SHALL be used for file delivery via the interfaces, subject to the following conditions:

- The interfaces using HTTP 1.1 [RFC 2616] SHALL support gzip, compress, deflate and identity content codings. Other content codings MAY be supported.
- The interfaces using HTTP 1.1 [RFC 2616] MAY use persistent connections, pipelining and chunked transfer coding.

5.4.1.2 Back-end Interface Messages

Messages to and from the FA or the FD are transported using HTTP as the transport by placing both the requests and the responses addressed to FA or FD into the payload of the HTTP messages. The requests SHOULD be transported using

HTTP POST and the responses SHOULD be transported using the HTTP responses corresponding to the HTTP POST requests. The syntax for the requests SHOULD be as follows:

- POST <host>/oma/bcast/fd HTTP/1.1\r\n<SessionCreationMessage>
- POST <host>/oma/bcast/fd HTTP/1.1\r\n<SessionDeletionMessage>
- POST <host>/oma/bcast/fd HTTP/1.1\r\n<FileInsertionMessage>
- POST <host>/oma/bcast/fd HTTP/1.1\r\n<FileRemovalMessage>

where the <host> denotes the part of the URI representing the address of the host.

Both the HTTP POST message and the corresponding HTTP response MAY also contain the following HTTP header fields:

- 'Content-Length',
- 'Content-Type' which if used SHALL be set to "text/xml" and
- 'Host' in case the 'Request-URI' is not in the absolute form specified in [RFC 2616].

5.4.1.2.1 Processing and Responding

The processing of the messages to the FA or the FD involves first the HTTP transport level to deliver the message from the CC to the FA or from the FA or FD. This is followed by the HTTP level passing the embedded XML message to the FA or FD. While the status and error codes corresponding to the processing in the HTTP level are signaled using the HTTP headers, the result of the FA or FD processing the XML request in the HTTP payload is signaled using XML messages placed into the payloads of the HTTP responses corresponding to the HTTP requests carrying the XML requests. Whenever an HTTP response contains an XML response from the FA or FD, the HTTP status code SHALL be set to 200 OK regardless of the contents of the XML response.

5.4.1.2.2 Session Creation

Session creation will be used for requesting the creation of session for file delivery. The parameters of the session can be assigned by either the network entity requesting session creation or the network entity being requested to create a session. When requesting the creation of a session, the CC or FA SHALL use the following 'SessionCreation' XML message in the HTTP payload.

Name	Type	Category	Cardinality	Description	Data Type
SessionCreation	E			Specifies the Session Creation Message. Contains the following attributes: tsi ipAddress portNumber useFDT bandwidth startTime endTime	

				blockLengthMax encodingSymbolLength useFEC fecCodeRate	
tsi	A	O	0..1	Transport Session Identifier of ALC/FLUTE	unsignedInt
ipAddress	A	O	0..1	Target IP Address of ALC/FLUTE	string
portNumber	A	O	0..1	Port number of Target Address of ALC/FLUTE	unsignedShort
useFDT	A	O	0..1	If this attribute set TRUE, it means the session will contain FDT Instances otherwise the session will have no FDT Instances.	boolean
bandwidth	A	O	0..1	Recommended bandwidth (bps) to the client backend entity. If this attribute is not present, it means all bandwidth and all capacity will be requested to reserve.	unsignedInt
startTime	A	M	1	The first moment of the Session to be created. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment of the Session to be created. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the “Session Deletion” request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
blockLengthMax	A	O	0..1	The maximum number of source symbols per source block	unsignedInt
encodingSymbolLength	A	O	0..1	The length of encoding symbol in bytes	unsignedInt
useFEC	A	O	0..1	Indicates whether the requested file delivery session is to use FEC	boolean
fecCodeRate	A	O	0..1	If FEC is used in the file delivery session, this attributes specifies the FEC code rate	decimal

Table 2: Request Message for Session Creation

Upon successful processing of the ‘SessionCreation’ request, the FA or FD SHALL use the following ‘SessionCreationRes’ message in HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type

SessionCreationRes	E			Specifies the response message to a session creation request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following elements: SessionInfo Parameter	
SessionInfo	E1	M	1	Specifies the created session information Contains the following attributes: tsi ipAddress portNumber useFDT bandwidth startTime endTime blockLengthMax encodingSymbolLength	
tsi	A	M	1	Transport Session Identifier of ALC/FLUTE	unsignedInt
ipAddress	A	M	1	Target IP Address of ALC/FLUTE	string
portNumber	A	M	1	Port number of Target Address of ALC/FLUTE	unsignedShort
useFDT	A	M	1	If this attribute set TRUE, it means the session will contain FDT Instances otherwise the session will have no FDT Instances.	boolean
bandwidth	A	M	0..1	This is the bandwidth assigned for the session.	unsignedInt
startTime	A	M	1	The first moment of the Session to be created. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment of the Session to be created. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the “Session Deletion” request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt

blockLengthMax	A	M	1	The maximum number of source symbols per source block	unsignedInt
encodingSymbolLength	A	M	1	The length of encoding symbol in bytes	unsignedInt
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 3: Response Message for Session Creation

5.4.1.2.3 Session Deletion

Session Deletion will be used for request to delete session for file delivery from CC to FA or from FA to FD.

When requesting a deletion of a session, clients of the FA or FD SHALL use the following XML message in the HTTP payload.

Name	Type	Category	Cardinality	Description	Data Type
SessionDeletion	E			Specifies the Session Deletion Message. Contains the following attributes: tsi ipAddress portNumber endTime	
tsi	A	M	1	Transport Session Identifier of ALC/FLUTE	unsignedInt

ipAddress	A	M	1	Target IP Address of ALC/FLUTE	string
portNumber	A	M	1	Port number of Target Address of ALC/FLUTE	unsignedShort
endTime	A	M	1	The moment the Session is to be deleted. If the value is zero, the transmission is to be ended immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt

Table 4: Request Message for Session Deletion

Upon successful processing of the 'SessionDeletion' request, the FA or FD SHALL use the following 'SessionDeletionRes' message in the HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type
SessionDeletionRes	E			Specifies the response message to a session deletion request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following elements: Parameter	
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 5: Response Message for Session Deletion

5.4.1.2.4 File Insertion

When requesting an insertion of a file into a file delivery session, the FA or FD SHALL use the following as the payload of the corresponding HTTP message; first the XML message 'FileInsertion' described below immediately followed by the payload of the file itself.

Name	Type	Category	Cardinality	Description	Data Type
FileInsertion	E			Specifies the File Insertion Message. Contains the following attributes: tsi ipAddress portNumber startTime endTime Contains the following element: File	
tsi	A	M	1	Transport Session Identifier of ALC/FLUTE	unsignedInt
ipAddress	A	M	1	Target IP Address of ALC/FLUTE	string
portNumber	A	M	1	Port number of Target Address of ALC/FLUTE	unsignedShort
startTime	A	M	1	The first moment the file is to be transmitted. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment the file is to be transmitted. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the "File Insertion" request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
File	E1	M	1	Element containing the metadata for the file to be inserted as specified by the FLUTE FDT File element [RFC 3926].	ComplexType

Table 6: Request Message for File Insertion

Upon successful processing of the 'FileInsertion' request message, the FA or FD SHALL use the following 'FileInsertionRes' message in the HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type

FileInsertionRes	E			Specifies the response message to a file insertion request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following elements: FileInformation Parameter	
FileInformation	E1	M	1	Contains the File Information Contains the following attributes: tsi ipAddress portNumber startTime endTime Contains the following element: File	
tsi	A	M	1	Transport Session Identifier of ALC/FLUTE	unsignedInt
ipAddress	A	M	1	Target IP Address of ALC/FLUTE	string
portNumber	A	M	1	Port number of Target Address of ALC/FLUTE	unsignedShort
startTime	A	M	1	The first moment the file is to be transmitted. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment the file is to be transmitted. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the “File Insertion” request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
File	E2	M	1	Element containing the metadata for the file to be inserted as specified by the FLUTE FDT File element [RFC 3926].	ComplexType

Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 7: Response Message for File Insertion

5.4.1.2.5 File Removal

The FA or FD MAY request the removal of files from a file delivery session. When requesting the removal of a file from a file delivery session, the FA or FD SHALL use the following 'FileRemoval' message as the payload of the corresponding HTTP message.

Name	Type	Category	Cardinality	Description	Data Type
FileRemoval	E			Specifies the File Removal Message. Contains the following attributes: toi tsi ipAddress portNumber endTime	
toi	A	M	1	Transport Object Identifier of file on the ALC/FLUTE session	unsignedInt
tsi	A	M	1	Transport Session Identifier of ALC/FLUTE	unsignedInt
ipAddress	A	M	1	Target IP Address of ALC/FLUTE	string

portNumber	A	M	1	Port number of Target Address of ALC/FLUTE	unsignedShort
endTime	A	M	1	The moment the file is to be removed from the file delivery session. If the value is zero, the transmission is to be ended immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt

Table 8: Request Message for File Removal

Upon successful processing of the 'FileRemoval' request message, the FA or FD SHALL use the following 'FileRemovalRes' message in the HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type
FileRemovalRes	E			Specifies the response message to a file removal request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following elements: Parameter	
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 9: Response Message for File Removal

5.4.2 Interface FD-B1

Interface FD-B1 provides the attributes of files as well as the files themselves for BCAST file distribution services. Files are provided by the BSDA to the BDS-SD within the BDS network and are distributed by the BDS-SD using a BDS specific file distribution mechanism.

This interface MAY be used to perform operations including, but not limited to:

- Distribution of content files
- Distribution of ROs as part of service or content protection
- Distribution of SGDUs and SGDDs as part of Service Guide delivery

5.4.2.1 Protocol Stacks

5.4.2.1.1 FD-B1

FD-B1 is the interface between BSD/A and BDS for file distribution. This interface is defined in each of the adaptation specifications, if applicable [BCAST13-BCMCS-Adaptation], [BCAST13-IPDC-DVBH-Adaptation], [BCAST13-MBMS-Adaptation]], [BCAST13-DVBSH-IPDC-Adaptation], [BCAST13-FLO-Adaptation], [BCAST13-WiMAX-Adaptation], [BCAST13-DVBNGH-Adaptation].

5.5 File Distribution over Interaction Channel

For file distribution over the Interaction Channel, two methods are specified:

- Use of FLUTE (section 5.5.1) and
- Use of HTTP/1.1 (section 5.5.2).

Terminals and networks that support the Interaction Channel SHALL support at least one method for file distribution over the Interaction Channel. Further, terminals that support FLUTE SHOULD support the use of FLUTE for file distribution over the Interaction Channel and terminals that support HTTP/1.1 SHALL support the use of HTTP/1.1 for file distribution over the Interaction Channel.

File distribution over the Interaction Channel is identical to the file repair procedure (section 5.3.3). Terminals and network MAY use FLUTE or MAY use HTTP/1.1 for file distribution over the Interaction Channel.

The Interaction Channel is used also for reception reporting and file repair purposes when used in conjunction with file distribution over the Broadcast Channel. Those aspects are not specified in this section but in section 5.3.

5.5.1 Use of FLUTE for File Distribution over Interaction Channel

The use of FLUTE is useful, for example, when the server initiates data delivery, i.e. pushes data to the device, and to enable service continuity for services delivered using FLUTE over the broadcast channel, when switching to the Interaction Channel.

A FLUTE session over the Interaction Channel SHALL be announced in an Access Fragment by instantiating an ‘AccessType’ element containing a ‘UnicastServiceDelivery’ child element with ‘type’ attribute set to “6” .

5.5.1.1 FLUTE Session Setup and Control with RTSP

SDP handling

The FLUTE specific SDP extensions are defined in [3GPP TS 26.346], chapter 7. For the FLUTE session establishment using RTSP, a control URI as defined in [RFC 2326] SHALL be present for the FLUTE media description. A control URI is defined by the “a=control:” SDP field according to [RFC 2326].

RTSP SETUP Method

The control URI as defined in [RFC 2326] 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 [RFC 2326] SHALL be "FLUTE/UDP". One and only one UDP port is allocated for each FLUTE channel. If the 'flute-ch' attribute is present for the FLUTE session, an according number of FLUTE sessions is established.

The following RTSP 1.0 defined 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.

RTSP PLAY Method

The PLAY method tells the server to start sending data, including FLUTE session data, as defined in [RFC 2326]. The RTSP server forwards the FLUTE packets as specified by the RTSP range header in the RTSP PLAY.

RTSP PAUSE Method

The PAUSE request causes the stream delivery, including all FLUTE sessions, to be interrupted (halted) as defined in [RFC 2326].

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 [RFC 2326].

5.5.2 Use of HTTP for File Distribution over Interaction Channel

When HTTP/1.1 is used for file distribution over the Interaction Channel, the HTTP URL of the resource according to [RFC 2616] SHALL be constructed from information declared in the Service Guide as defined below (see also sections 5.1.2.4 and 5.1.2.2 of [BCAST13-SG]). The conversion between that HTTP URL and the HTTP Request-URI SHALL be done in accordance with [RFC 2616], section 5.1.2.

The 'http', 'host' and (if given) 'port' components of the HTTP URL SHALL always be taken from the 'AccessServerURL' element in the 'Access' fragment. The information to fill the 'abs_path' and 'query' components of the HTTP URL MAY be signaled either in the 'AccessServerURL' element in the 'Access' fragment or in the 'contentLocation' attribute in the 'Schedule' fragment or in both places. Note that 'AccessServerURL' is an absolute URL and 'contentLocation' a relative one which can override and/or complement information in 'AccessServerURL'. This implies that 'AccessServerURL' MUST comply with the syntax of an absolute HTTP URL defined in [RFC 2616]. The components in the HTTP URL SHALL be filled as follows:

- in case 'abs_path' is not signaled in 'AccessServerURL' but 'query' is signaled there, then an illegal combination has occurred
- in case 'abs_path' is signaled in neither 'AccessServerURL' nor 'contentLocation' but 'query' is signaled in 'contentLocation', then an illegal combination has occurred
- in case the 'abs_path' component is signaled in neither 'AccessServerURL' nor 'contentLocation', the string "/" SHALL be used as the 'abs_path' component of the resulting HTTP URL, as defined in [RFC 2616]
- in case the 'abs_path' component is signaled in 'AccessServerURL' only, this information SHALL be used as the 'abs_path' component of the resulting HTTP URL, and
    - in case the 'query' component is signaled in either 'AccessServerURL' or 'contentLocation' but not in both, this information SHALL be used as the 'query' component of the resulting HTTP URL

    - in case the ‘query’ component is signaled in both ‘AccessServerURL’ and ‘contentLocation’, the information in ‘contentLocation’ SHALL be used as the ‘query’ component of the resulting HTTP URL

- in case the ‘abs_path’ component is signaled in ‘contentLocation’ only, this information SHALL be used as the ‘abs_path’ component of the resulting HTTP URL, and
    - in case the ‘query’ component is signaled in ‘contentLocation’ only, this information SHALL be used as the ‘query’ component of the resulting HTTP URL
- in case the ‘abs_path’ component is signaled in both ‘AccessServerURL’ and ‘contentLocation’, the information in ‘contentLocation’ SHALL be used as the ‘abs_path’ component of the resulting HTTP URL, and
    - in case the ‘query’ component is signaled in ‘contentLocation’, this information SHALL be used as the ‘query’ component of the resulting HTTP URL, regardless whether or not the ‘query’ component is also signaled in ‘AccessServerURL’

The table below lists all the possible combinations.

AccessServerURL carries		**contentLocation carries**		**Resulting HTTP URL components**	
abs_path	**query**	**abs_path**	**query**	**abs_path**	**query**
-	-	-	-	“/”	-
-	-	-	C	illegal combination	
-	-	C	-	C	-
-	-	C	C	C	C
-	A	-	-	illegal combination	
-	A	-	C	illegal combination	
-	A	C	-	illegal combination	
-	A	C	C	illegal combination	
A	-	-	-	A	-
A	-	-	C	A	C
A	-	C	-	C	-
A	-	C	C	C	C
A	A	-	-	A	A
A	A	-	C	A	C
A	A	C	-	C	-
A	A	C	C	C	C

Legend: “A” – component carried in AccessServerURL, “C” – component carried in contentLocation, “-“ – component not carried

In the following, an example of the file request is given:

GET /news/latest.txt **HTTP/1.1**

HOST: www.example.com

The HTTP messages related to the associated delivery procedures such as reception reporting and file repair are specified in section 5.3.

5.6 File Distribution over Hybrid Broadcast/Interaction Channel

Multiple file delivery sessions form a hybrid file download service in case they are combined and deliver the same transport objects over both Interaction Channel and Broadcast Channel. For such file delivery sessions the network SHALL synchronize FDT Instances and transport objects between the file delivery sessions. More specifically, the following applies for file delivery sessions constituting a hybrid file download service:

- The network SHALL deliver the same transport objects in all file delivery sessions and the terminal SHOULD use received file fragments across file delivery sessions.
- The network SHALL align the FDT Instances delivered in all file delivery sessions and the terminal SHOULD use received FDT Instances across file delivery sessions.
- The parameters listed in section 5.2.6.4 which applies to both FDT-Instance level information and all files of a FLUTE session SHALL be identical in all file delivery sessions.
- The terminal SHOULD interpret associated delivery procedure descriptions received within one file delivery session as applicable for all file delivery sessions.

For file delivery sessions forming a hybrid file download service, the terminal SHALL NOT assume any of the session descriptor parameters listed in section 5.1.2.5.3.1 of [BCAST13-SG] to be identical.

A hybrid file download service is signaled by multiple ‘Access’ fragments referencing the same ‘Service’ or ‘Content’ fragment as described in section 5.8.1.1 of [BCAST13-SG]. See section I.3.3 in [BCAST13-SG] for a Service Guide instantiation example which describes a hybrid file download service by having more than one ‘Access’ fragment referencing the same ‘Service’ fragment.

6. Stream Distribution

6.1 Introduction

The purpose of the BCAST Stream Distribution Function is to deliver stream services over IP. To facilitate stream distribution, RTP SHALL be used.

The specification for the OMA BCAST Stream Distribution function consists of the specification of four interfaces: SD-1, SD-2, SD-5 and SD-6. The interfaces SD-5 and SD-6 are terminal-network interfaces and the functional entities across these interfaces are the Stream Delivery Client Component (SDC) in the terminal and the Stream Delivery Component (SD) in the network. These interfaces are specified in sections 6.2, 6.3 and 6.5. The interfaces SD-1 and SD-2 are back-end interfaces within the system(s) serving the OMA Mobile Broadcast Services and the functional entities across these interfaces are the Stream Distribution Component (SD) and the Stream Application Component (SA), both in the network. These back-end interfaces are specified in section 6.4.

6.2 RTP as Stream Transport Protocol

The Real-Time Transport Protocol (RTP) [RFC 3550] is a protocol used for unreliable delivery of streams. RTP provides means for sending real-time or streaming data over UDP. The transmission of RTP Control Protocol (RTCP) packets in the downlink (sender reports) is mandatory for synchronizing multiple RTP streams. The transmission of RTCP packets in the uplink (receiver reports) is not allowed. RTCP receiver reports SHALL be turned off by SDP RR bandwidth modifiers.

Over the interface SD-5 the following specification applies:

- Both the server and the terminal SHALL support RTP for the delivery of streams.
- The sender SHALL send RTCP sender reports for all sessions involving more than a single RTP stream. In case the service is made of a single RTP stream it is not mandatory to send RTCP sender reports.
- The terminal SHALL support receiving RTCP packets (sender reports).
- The sender SHALL turn off RTCP receiver reports using signalling in the SDP session description.

Over the interface SD-6 the following specification applies:

- If the sender has signalled that no RTCP receiver reports SHALL be sent, the terminal MUST NOT send RTCP packets (receiver reports).

6.2.1 RTP Payload Formats

RTP can, in general, transport any audio/visual data, if an RTP payload format is defined for it. However, the BCAST enabler does not specify media codecs and, consequently, RTP payload formats are not be defined in this specification. The BCAST adaptation specifications [BCAST13-MBMS-Adaptation], [BCAST13-BCMCS-Adaptation], [BCAST13-IPDC-DVBH-Adaptation]], [BCAST13-DVBSH-IPDC-Adaptation], [BCAST13-FLO-Adaptation], [BCAST13-WiMAX-Adaptation], [BCAST13-DVBNGH-Adaptation]and [BCAST13-DVBNGH-Adaptation] contain specification text, purely in the context of making external reference, on the use of codecs and associated RTP payload formats for a codec defined by the BCAST enabler in conjunction with the respective standards organization BDS.

6.2.2 Forward Error Correction

The FEC Raptor scheme (FEC encoding ID 1) MAY be supported. The FEC Raptor scheme is specified in [3GPP TS 26.346] for MBMS.

6.2.3 Buffer Control for Stream Distribution

Due to the variable bit rate nature of some media streams (especially video streams), initial buffering at the receiver becomes necessary. 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>"

The 'X-initpredecbufperiod' attribute shall be used as defined in [3GPP TS 26.234], section 5.3.3.2. This parameter specifies the removal delay of the first access unit expressed in RTP clock rate. The decoder buffer size SHOULD be set according to the media profile and level requirements.

For various reasons, a BDS may require buffer control for streaming delivery. While being out of scope of this specification, these reasons might have an impact on streaming delivery in BCAST as well. When this happens, the buffer control mechanism has to be taken into account in BCAST. Details for this can be found in the corresponding adaptation specification.

6.3 Associated Procedures for Stream Distribution

An associated procedure description instance (configuration information) for the streaming associated delivery procedure MAY be delivered to BCAST Terminals as follows:

- In the Service Guide prior to the BCAST stream delivery session along with the session description (out-of-band of that session); or
- In-band within a BCAST stream delivery session.

The most recently delivered configuration file (i.e. the one with the highest version number) 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 streaming 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. In the Service Guide, the associated procedure description instance is clearly identified using a URI, to enable the SD-C to cross-reference out-of-band configuration files.

The MIME application type "application/vnd.oma.bcast.associated-procedure-parameter" identifies associated delivery procedure description instances (configuration files).

In XML, each associated delivery procedure entry shall be configured using a 'StreamingAssociatedProcedure' element. All configuration parameters of one associated delivery procedure are contained as attributes of a 'StreamingAssociatedProcedure' element. The associated delivery procedure description is specified formally as an XML schema/table in section 6.3.1.

Following the reception of streaming content, a reception reporting procedure MAY be initiated by the SD-C to the SD.

For BCAST streaming delivery, the reception reporting procedure is used to report statistics on the stream.

If the SD provided parameters requiring reception reporting confirmation then the SD-C SHALL confirm the content reception.

Transport errors can prevent a SD-C 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. The SD-C SHALL behave according to the information it has using the following procedure:

1. Identifies the reception of streaming content and commences statistics gathering based upon the streaming associated procedure.
2. Selects a time (random time) at which a reception report will be sent and selects a SD from a list - both randomly and uniformly distributed.
3. Sends a reception report message to the selected SD at the selected time.

The back-off timer used in section 5.3.3.3 SHALL be used by the terminal when selecting a time to send the reception report.

A list of report server URIs is provided by a list of server URIs as elements of the streaming associated procedure description in the 'StreamingAssociatedProcedure' element. Server URIs host identity may also be given as IP addresses. The report server URIs of a single associated procedure description shall be of the same type, e.g. all IP addresses of the same version, or all domain names. The SD-C randomly selects one of the server URIs from the list, with uniform distribution.

Support for the streaming associated procedure description and the streaming reception reporting is OPTIONAL for the network as well as for the terminal.

6.3.1 Associated Procedure Description

A stream reception reporting procedure is requested of the SD-C by the SD via the Service Guide. Compared to reception reporting of file delivery sessions in section 5.3.2, where the reception confirmation units are clear (i.e. files), reception reporting for streaming sessions by its nature requires measurement periods to be defined to create a measurement unit. For collecting reception statistics of streaming sessions the following six methods and respective measurement periods are defined:

- Session based Measurements: Packet loss measurements are conducted for a full duration of the related session and then reported.
- Fixed-duration based Measurements: Packet loss measurements are conducted for a fixed duration and then reported. The duration is predefined with start and end times represented in RTP timestamps.
- Interval based Measurements: Packet loss measurements are continuously conducted and reported in predefined fixed intervals. The fixed interval is defined by an interval value to be calculated relative to RTP timestamps.
- Threshold based Measurements: Packet loss measurements are continuously conducted and reported whenever a predefined threshold is reached. The threshold value is compared to the reception rate of the stream.
- Event-triggered Measurements: Packet loss measurements are continuously conducted and reported whenever a predefined threshold is reached. The trigger value is compared to the reception rate of the stream.
- MBMS based Measurements: Measurements are conducted upon 3GPP MBMS associated delivery procedures for streaming content. 3GPP MBMS associated delivery procedure description elements and attributes should be used as defined in [3GPP TS 26.346].

A stream reception reporting procedure is delivered by the SD and requested of the SD-C using a 'StreamingAssociatedProcedure' element. The XML syntax for the 'StreamReceptionReport' element is summarized in the table below.

Name	Type	Category	Cardinality	Description	Data Type
StreamingAssociatedProcedure	E			Associated delivery procedure for stream delivery Contains the following attributes: offsetTime randomTimePeriod Contains the following elements: ServerURI MeasurementType MBMSMeaurement	

offsetTime	A	O	0..1	The suppression time to wait before requesting repair. Refer to 5.3.3.4 for details.	unsignedLong
randomTimePeriod	A	O	0..1	The time window length over which a SD-C SHALL calculate a *random time*. Refer to 5.3.3.4 for details.	unsignedLong
ServerURI	E1	M	1..N	The return address for the reception report.	anyURI
MeasurementType	E1	O	0..1	This element defines the type of measurement and reception report that should be executed by the terminal. Contains the following elements: SessionMeasurement FixedDurationMeasurement IntervalMeasurement ThresholdMeasurement EventTriggeredMeasurement	
SessionMeasurement	E2	O	0..1	This element requests the terminal to send a reception report for the packet loss measurement of a complete session.	boolean
FixedDurationMeasurement	E2	O	0..1	This element requests the terminal to send a reception report for the packet loss measurement of a fixed duration of time as defined by the attributes. Contains the following attributes: startRTPTimestamp endRTPTimestamp	boolean
startRTPTimestamp	A	M	1	The RTP timestamp used to begin the measurement.	unsignedInt
endRTPTimestamp	A	M	1	The RTP timestamp used to finish the measurement.	unsignedInt
IntervalMeasurement	E2	O	0..1	This element requests the terminal to send periodic reception reports for the packet loss measurement on a fixed interval basis. Contains the following attributes: interval	boolean
interval	A	M	1	The interval at which the terminal should send reception reports. The start of the interval is calculated based on the moment the terminal receives the first RTP packet.	unsignedInt

ThresholdMeasurement	E2	O	0..1	This element requests the terminal to send a reception report for the packet loss measurements whenever the packet loss is greater than the designated threshold. Contains the following attributes: threshold	boolean
threshold	A	M	1	The threshold value which the terminal should use to check whether if it should send reception reports. The terminal will begin measurements when it receives the first packet but will only send reception reports when the threshold value is breached.	float
EventTriggeredMeasurement	E2	O	0..1	This element requests the terminal to send a reception report for packet loss measurement after an event is triggered. Contains the following attributes: trigger	boolean
trigger	A	M	1	The trigger value which the terminal should use to check whether if it should start to create data for a reception report. The terminal will begin measurements when it receives the first packet but will only send a reception report with data from when the trigger value was reached onwards.	float
MBMSMeasurement	E1	O	0..1	3GPP MBMS based reception reporting. The associated delivery procedure description for streaming reception should be inserted here. Refer to [3GPP TS 26.346] for details.	anyType

Table 10: XML Syntax for Stream Associated Delivery Procedure Description

6.3.2 Stream Reception Report

A stream reception report is sent to the server based upon the measurements made according to the original request in the 'StreamingAssociatedProcedure' element. A stream reception report is delivered using a 'StreamingReceptionReport' element. Depending upon the type of measurement conducted, one or more reception reports may be delivered to the server. The XML syntax for the 'StreamReceptionReport' element can be found in [BCAST10-XMLSchema-SD-ReceptionReport] and is summarized in the table below.

Name	Type	Category	Cardinality	Description	Data Type
StreamingReceptionReport	E			Reception report for stream delivery Contains the following attributes: serverURI globalServiceID Contains the following elements: DeviceID SessionID MBMSMetrics	
globalServiceID	A	M	1	The identifier of the service that the measurement was performed on.	anyURI
DeviceID	E1	M	1	A unique device identification known to the BSM Contains the following attribute: type	unsignedInt
type	A	M	1	Specifies the type of device ID. The following values are allowed: 0 –DVB Device ID 1 –3GPP Device ID (IMEI) 2 –3GPP2 Device ID (MEID) 3-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
Session	E1	M	1..N	Information identifiying the session and related data that was monitored for recpetion reporting. Contains the following attributes: id Contains the following elements: Content	
id	A	M	1	The identifier of the session that was monitored for reception reporting, originating from the o= line in the SDP description of the streaming session.	anyURI

Content	E2	M	1..N	Contains the ID and the measurement results for the content on which the measurement was performed. Contains the following attributes: globalContentID reportType measurementStartRTPTimestamp measurementEndRTPTimestamp expectedTotalPacket receivedTotalPackets lostTotalPackets receptionRatio serviceArea cellID	
globalContentID	A	M	1	The identifier of the content, globally unique	anyURI
reportType	A	M	1	The type of the reception report The following values are specified: 0 – SessionMeasurement 1 – FixedDurationMeasurement 2 – IntervalMeasurement 3 – ThresholdCheckingMeasurement 4 – EventTriggeredMeasurement 5 – MBMSMeasurement 6 - 127 reserved for future use 128 - 255 reserved for proprietary use	unsingedByte
measurementStartRTPTimestamp	A	M	1	The RTP timestamp used for the start time of the measurement.	unsignedInt
measurementEndRTPTimestamp	A	M	1	The RTP timestamp used for the end time of the measurement.	unsignedInt
expectedTotalPackets	A	O	0..1	The total number of expected packets during the measurement period.	unsignedInt
receivedTotalPackets	A	O	0..1	The total number of successfully received packets during the measurement period.	unsignedInt
lostTotalPackets	A	O	0..1	The total number of packets lost during the measurement period.	unsignedInt

receptionRatio	A	M	1	The reception ratio during the measurement period. Calculated from expectedTotalPackets and receivedTotalPackets.	Float
serviceArea	A	O	0..1	The area in which the measurements were taken.	unsignedInt
cellID	A	O	0..1	The cell in which the measurements were taken. Note: Only applies to 3GPP/2	unsignedInt
MBMSmetrics	E1	O	0..1	3GPP MBMS based reception reporting. The associated delivery procedure reception report for streaming reception should be inserted here. Refer to [3GPP TS 26.346] for details	anyType

Table 11: XML Syntax for Streaming Reception Report

6.3.3 Protocols

HTTP 1.1 over TCP/IP SHALL be used for associated streaming procedures. The HTTP POST command SHALL be used to deliver the streaming reception report request message and streaming reception report message defined in 6.3.1 and 6.3.2.

6.3.4 XML Schema for Associated Streaming Procedures

The XML syntax for the streaming associated delivery procedure description can be found in [BCAST10-XMLSchema-SD-AssociatedProcedure].

The XML syntax for the stream reception report can be found in [BCAST10-XMLSchema-SD-ReceptionReport].

6.4 Stream Distribution over Back-end Interfaces

This section specifies interfaces between logical BCAST “back-end” entities. The specification is applicable if the interfaces are exposed in a BCAST implementation. If a BCAST implementation does not expose the interfaces, i.e, they are internal to that implementation, they MAY be realized using protocols and methods not specified here.

The following two types of delivery are possible for stream distribution:

- Non-live stream delivery - SA delivers streams which are stored in files which SHALL be streamed to end-users by the SD when required.
- Live stream delivery - Live streams are provided from the CC to the end-user. The SA receives the live stream from the CC and SHALL forward the stream to the SD for streaming to users.

6.4.1 Interfaces SD-1 and SD-2 for Non-live Streaming

Interface SD-1 between the CC and SA provides the attributes of non-live stream files as well as the stream files themselves for BCAST stream distribution services.

Interface SD-2 between the SA and SD provides the attributes of non-live stream files as well as the stream files themselves for BCAST stream distribution services.

6.4.1.1 Protocol Stacks

The protocol stack shown in Figure 3 SHALL be used for non-live stream delivery over backend interfaces SD-1 between CC and SA and SD-2 between SA and SD. For secure operation over these backend interfaces, the entities implementing the interface SHALL support HTTPS, where HTTPS SHALL be based on SSL3.0 [SSL30] and TLS 1.0 [RF C2246].

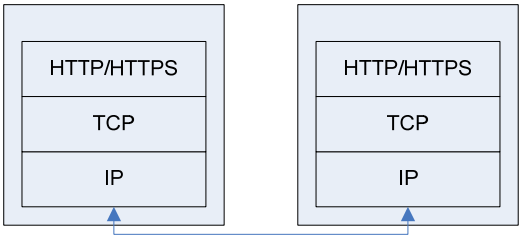

Figure 3: Protocol Stack for Back-end Interface for Stream Delivery

HTTP 1.1 over TCP/IP SHALL be used for file delivery via the interfaces, subject to the following conditions:

- The interfaces using HTTP 1.1 [RFC 2616] SHALL support gzip, compress, deflate and identity content codings. Other content codings MAY be supported.
- The interfaces using HTTP 1.1 [RFC 2616] MAY use persistent connections, pipelining and chunked transfer coding.

6.4.1.2 Back-end Interface Messages

Messages to and from the SA or SD are transported using HTTP as the transport by placing both the requests and the responses addressed to SA or SD into the payload of the HTTP messages. The requests SHOULD be transported using HTTP POST and the responses SHOULD be transported using the HTTP responses corresponding to the HTTP POST requests. The syntax for the requests SHOULD be as follows.

- POST <host>/oma/bcast/sd HTTP/1.1\r\n<StreamSessionCreationMessage>
- POST <host>/oma/bcast/sd HTTP/1.1\r\n<StreamSessionDeletionMessage>
- POST <host>/oma/bcast/sd HTTP/1.1\r\n<StreamInsertionMessage>
- POST <host>/oma/bcast/sd HTTP/1.1\r\n<StreamRemovalMessage>

where the <host> denotes the part of the URI representing the address of the host.

Both the HTTP POST message and the corresponding HTTP response MAY also contain the following HTTP header fields:

- 'Content-Length',
- 'Content-Type' which if used SHALL be set to "text/xml" and
- 'Host' in case the 'Request-URI' is not in the absolute form specified in [RFC 2616].

6.4.1.2.1 Processing and Responding

The processing of the message to the SA or SD involves first the HTTP transport level to deliver the message from the CC to the SA or from the SA or SD. This is followed by the HTTP level passing the embedded XML message to the SA or SD. While the status and error codes corresponding to the processing in the HTTP level are signaled using the HTTP headers, the result of the SA or SD processing the XML request in the HTTP payload is signaled using XML messages placed into the payloads of the HTTP responses corresponding to the HTTP requests carrying the XML requests. Whenever a HTTP response contains an XML response from the FA or FD, the HTTP status code SHALL be set to 200 OK regardless of the contents of the XML response.

6.4.1.2.2 Stream Session Creation

Stream session creation will be used for requesting the creation of session for stream file delivery. The parameters of the session can be assigned by either the network entity requesting session creation or the network entity being requested to

create a session. When requesting the creation of a session, CC or SA SHALL use the following 'StreamSessionCreation' XML message in the HTTP payload.

Name	Type	Category	Cardinality	Description	Data Type
StreamSession Creation	E			Specifies the Session Creation Message. Contains the following attributes: startTime endTime datarate fecRate Contains the following element: SDP	
startTime	A	M	1	The first moment of the Session to be created. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment of the Session to be created. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the "Session Deletion" request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
datarate	A	M	1	Required data rate Note: As this shall be streamed to end-users reception time must be taken into account.	unsignedInt
fecRate	A	O	0..1	Required FEC code rate expressed as a decimal number between 0 and 1	decimal
SDP	E1	M	1	A session description in SDP (IETF session description protocol) format	string

Table 12: Request Message for Stream Session Creation

Upon successful processing of the 'StreamSessionCreation' request message, the SA or SD SHALL use the following 'StreamSessionCreationRes' message in HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type
StreamSession CreationRes	E			Specifies the Response message for Session Creation Request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following attributes: sessionID Contains the following elements: SessionInfo Parameter	
sessionID	A	M	1	The assigned session identifier	unsignedInt
SessionInfo	E1	M	1	Specifies the created session information. Contains the following attributes: startTime endTime fecRate datarate Contains the following elements: SDP	
startTime	A	O	0..1	The first moment of the Session to be created. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	O	0..1	The last moment of the Session to be created. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the “Session Deletion” request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
fecRate	A	O	0..1	Assigned FEC code rate	decimal
datarate	A	M	1	Required data rate Note: As this shall be streamed to end-users reception time must be taken into account.	unsignedInt

SDP	E2	O	0..1	A session description in SDP (IETF session description protocol) format. If this element is present, it signals that the SDP as signaled in the request message was modified and this element contains the modified SDP.	string
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	0..1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	0..1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 13: Response Message for Stream Session Creation

6.4.1.2.3 Session Deletion

Session Deletion will be used for request to delete session for stream delivery from CC to SA or from SA to SD.

When requesting the deletion of a session, clients of the SA or SD SHALL use the following XML message in the HTTP payload.

Name	Type	Category	Cardinality	Description	Data Type
StreamSessionDeletion	E			Specifies the Session Deletion Message Contains the following attributes: sessionID endTime	
sessionID	A	M	1	Session identifier	unsignedInt
endTime	A	M	1	The moment the session is to be deleted. If the value is zero, the transmission is to be ended immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt

Table 14: Request Message for Stream Session Deletion

Upon successful processing of the 'StreamSessionDeletion' request message, the FA or FD SHALL use the following 'StreamSessionDeletionRes' message in the HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type
StreamSession DeletionRes	E			Specifies the response message to a session deletion request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following attributes: sessionID Contains the following elements: Parameter	
sessionID	A	M	1	Session identifier	unsignedInt
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	0..1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	0..1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 15: Response Message for Stream Session Deletion

6.4.1.2.4 Stream Insertion

When requesting the insertion of a stream file into a stream delivery session, the SA or SD SHALL use the following as the payload of the corresponding HTTP message; first the XML message 'StreamInsertion' described below immediately followed by the payload of the file itself.

Name	Type	Category	Cardinality	Description	Data Type
StreamInsertion	E			Specifies the Stream Insertion Message. Contains the following attributes: sessionID startTime endTime Contains the following element: StreamInfo SDP	
sessionID	A	M	1	Session identifier, to which the stream file is to be inserted	unsignedInt
startTime	A	M	1	The first moment the stream is to be transmitted. If the value is zero, the transmission is to be started immediately. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment the stream is to be transmitted. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the "Stream Removal" request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
StreamInfo	E1	M	1..N	Specifies the parameters of a single media stream. Contains the following attributes: ipAddress portNumber	
ipAddress	A	M	1	Target IP Address of RTP	string
portNumber	A	M	1	Port number of target address of RTP	unsignedShort
SDP	E1	M	1	A session description in SDP (IETF session description protocol) format .	string

Table 16: Request Message for Stream Insertion

Upon successful processing of the 'StreamInsertion' request message, the FA or FD SHALL use the following 'StreamInsertionRes' message in the HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type
StreamInsertionRes	E			Specifies the response message to a stream insertion request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following attributes: sessionID streamID Contains the following elements: StreamInformation SDP Parameter	
sessionID	A	M	1	Session identifier	unsignedInt
streamID	A	M	1	Stream identifier unique in the scope of the session	unsignedInt
StreamInformation	E1	M	1..N	Contains the Stream Information Contains the following attributes: ipAddress portNumber startTime endTime Contains the following elements: SDP	
ipAddress	A	M	1	Target IP Address of RTP	string
portNumber	A	M	1	Port number of Target Address of RTP	unsignedShort

startTime	A	M	1	The first moment the stream is to be transmitted. If the value is zero, the transmission is to be started immediately. This field is expressed as the most significant 32 bits of NTP time format.	unsignedInt
endTime	A	M	1	The last moment the stream is to be transmitted. If the value is zero, the transmission is to be ended only when requested explicitly by the client using the "Stream Removal" request. This field expressed as the most significant 32 bits of NTP time format.	unsignedInt
SDP	E1	O	0..1	A session description in SDP (IETF session description protocol) format. If this element is present, it signals that the SDP as signaled in the request message was modified and this element contains the modified SDP.	string
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	0..1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string
reason	A	O	0..1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 17: Response Message for Stream Insertion

6.4.1.2.5 Stream Removal

The SA or SD MAY request removal of stream files from stream delivery sessions. When requesting the removal of a stream file from a stream delivery session, the SA or SD SHALL use the following 'StreamRemoval' message as the payload of the corresponding HTTP message.

Name	Type	Category	Cardinality	Description	Data Type
StreamRemoval	E			Specifies the Stream Removal Message Contains the following attributes: sessionID streamID endTime	
sessionID	A	M	1	Session identifier	unsignedInt
streamID	A	M	1	Stream identifier unique in the scope of the session	unsignedInt
endTime	A	M	1	The moment the stream is to be removed. If the value is zero, the transmission is to be ended immediately. This field is expressed as the most significant 32 bits of NTP time format.	unsignedInt

Table 18: Request Message for Stream Removal

Upon successful processing of the 'StreamRemoval' request message, the SA or SD SHALL use the following 'StreamRemovalRes' message in the HTTP payload of the response.

Name	Type	Category	Cardinality	Description	Data Type
StreamRemovalRes	E			Specifies the response message to a stream removal request. If an error occurs, then at least one Parameter element SHALL be present in the response. Contains the following elements: Parameter	
Parameter	E1	O	0..N	If this element is present, it signals that there was an error with a parameter in the request. Contains the following attributes: field reason	
field	A	O	0..1	Character string specifying the name of the faulty parameter in the request message. This is a name of an element or attribute in the request.	string

reason	A	O	0..1	Contains the reason for the rejection of the parameter, according to the global status code (as specified in [BCAST13-Services]). For instance, if the value is set to "17 – Information Element non-existent", it signals that the FA or FD could not find the parameter at all in the request. If the value is set to "021 – Information Invalid", it signals that although the parameter is present in the request, the value of the parameter is not accepted by the FA or FD.	unsignedByte

Table 19: Response Message for Stream Removal

6.4.2 Interface SD-B1

Interface SD-B1 is use to provide attributes of streaming contents in addition to the streaming contents for BCAST streaming services. Live streams are provided from the CC to the end-user. The SD receives the live stream from the SA and forwards the stream to the streaming distribution function within the BDS-SD. The stream MAY be service protected by the BSDA.

6.4.2.1 Protocol Stacks

6.4.2.1.1 SD-B1

SD-B1 is the interface between BSD/A and BDS for stream distribution. This interface is defined for each of the adaptation specifications, if applicable [BCAST13-BCMCS-Adaptation], [BCAST13-IPDC-DVBH-Adaptation], [BCAST-11-MBMS-Adaptation]], [BCAST13-DVBSH-IPDC-Adaptation], [BCAST13-FLO-Adaptation], [BCAST13-WiMAX-Adaptation], [BCAST13-DVBNGH-Adaptation].

6.5 Stream Distribution over Interaction Channel

Terminals and networks that support the Interaction Channel SHALL support stream distribution over the Interaction Channel.

Terminals that support stream distribution over the Interaction Channel SHOULD implement the streaming service as defined either in [3GPP TS 26.234] or in [3GPP2 C.S0046-0], with respect to the following:

- Transport of continuous media (video, audio, speech and timed text) using RTP/UDP/IP;
- Support of the RTP Control Protocol (RTCP) to report feedback;
- Use of the Real Time Streaming Protocol (RTSP) to set up and control the point-to-point stream session;
- Session Description Protocol (SDP) shall be used as the format of the presentation description for both the terminal and BSDA. In addition, SDP delivered to the client shall declare the media types to be used in the session using a codec-specific MIME media type appropriate for [3GPP TS 26.234] or [3GPP2 C.S0046-0];
- Use of HTTP/TCP/IP for the transport of certain media types which don't currently have corresponding RTP definitions (such as synthetic audio, Unicode text, bitmap graphics, still images, and vector graphics);
- ".3gp" or ".3g2" file formats when multiple media elements are aggregated for combined delivery over HTTP/TCP/IP;
- Capability exchange, to enable the streaming servers to provide a wide range of devices with content suitable for the particular client of concern.

The session establishment SHALL be based on an RTSP URL or an SDP identifying the streaming server and the content as defined in [RFC 2326]. In the Service Guide, the SDP SHALL be designated by the 'SDPRef' element or embedded in the

'SDP' element of the 'SessionDescription' element of the 'UnicastServiceDelivery' element in the 'Access' fragment. The RTSP Request-URI according to [RFC 2326] SHALL be constructed from information declared in the ServiceGuide as defined below (see also sections 5.1.2.4 and 5.1.2.2 of [BCAST13-SG]):

The 'rtsp', 'host' and (if given) 'port' components of the Request-URI SHALL always be taken from the 'AccessServerURL' element in the 'Access' fragment. The 'abs_path' component of the Request-URI MAY be signaled either in the 'AccessServerURL' element in the 'Access' fragment or in the 'contentLocation' attribute in the 'Schedule' fragment or in both places. Note that 'AccessServerURL' is an absolute URL and 'contentLocation' a relative one which can override and/or complement information in 'AccessServerURL'. The corresponding components in the RTSP Request-URI SHALL be filled as follows:

- in case the 'abs_path' component is signaled in neither 'AccessServerURL' nor 'contentLocation', the string "/" SHALL be used as the 'abs_path' component of the resulting 'Request-URI'
- in case the 'abs_path' component is signaled in either 'AccessServerURL' or 'contentLocation' but not in both, this information SHALL be used as the 'abs_path' component of the resulting 'Request-URI'
- in case the 'abs_path' component is signaled in both 'AccessServerURL' and 'contentLocation', the information from 'contentLocation' SHALL be used as the 'abs_path' component of the resulting 'Request-URI', overriding the information in 'AccessServerURL'

The terminal MAY use the RTSP range header field to request the unicast delivery to start from a given time point. The range SHALL be in NPT format and the time values SHALL be relative to the session start time as indicated in the session description.

NOTE: Specification or negotiation of ports used for unicast service delivery is handled by the unicast distribution mechanism used for delivering that service. For example, for RTSP and PSS based systems (values 3 and 4), port negotiation is done within the RTSP signalling exchange.

To improve user experience, there exist methods for fast switching between unicast streaming channels. BCAST Terminals MAY support the Fast Content Switching mechanism defined in [3GPP TS 26.234]. If the terminal contains an implementation of the Fast Content Switching mechanism according to [3GPP TS 26.234], that mechanism SHALL also be supported in BCAST mode of operation.

To further improve user experience, there exists a mechanism for time-shifting live streaming sessions. A server side time-shift buffer allows the terminal to pause live sessions and navigate (rewind, fast forward) in the offered time-shift buffer range. BCAST servers and terminals MAY support the time-shifting functionality defined in [3GPP TS 26.234].

6.5.1 Advisable Time Ranges for Access Switch

BCAST terminals MAY support retrieving information that describes positions in the content where a user is likely to perceive a disturbance as having a low impact on the viewing experience. The terminal can use this information, e.g., in a hybrid broadcast/unicast scenario to plan and execute a controlled switch from interaction access to broadcast access as described in section D.2.3. BCAST servers MAY support this functionality.

The RTSP GET_PARAMETER method as defined in [RFC 2326] SHALL be used to convey the advisable time ranges for access switch. In addition, the following applies:

- The RTSP GET_PARAMETER request SHALL be sent from terminal to server.
- The 'Content-Type' header SHALL be set to "text/parameters".
- Parameter "advisableswitchtime" SHALL be included in the body of the RTSP GET_PARAMETER request.
- The RTSP GET_PARAMETER response SHALL be sent from server to terminal.
- The syntax of the content returned in the RTSP GET_PARAMETER response body SHALL be defined in ABNF [RFC 4234] as follows:

  Advisable-Switch-Time = "advisableswitchtime" ":" [Range *(“,” Range)] CRLF

Range = "range" ":" Ranges-Specifier

Ranges-Specifier = as defined in [RFC 2326]

- If there are no suitable time ranges for access switch, the server SHALL respond without instantiating the 'Range' rule as enabled by the syntax of 'Advisable-Switch-Time'.
- The server SHALL use a 'Ranges-Specifier' with a time format which enables the terminal to map the provided time with corresponding RTP timestamp.

One possible source of advisable time range information is content and service providers which typically use play lists with program tags and time information. This information can be extracted, transformed into advisable time ranges for access switch, and made available to the BCAST server. However, the method used for determining advisable time range information and the signalling thereof at the network side is out of scope of this specification.

In the following example it is shown how the terminal successfully retrieves time ranges suitable for access switch.

C->S GET_PARAMETER rtsp://example.com/service1.3gp RTSP/1.0
Cseq: 122
Session: 17903320
Content-Type: text/parameters
Content-length: 21

advisableswitchtime

The response includes the suitable ranges:

S->C RTSP/1.0 200 OK
Cseq: 122
Session: 17903320
Content-Type: text/parameters
Content-Length: 159

advisableswitchtime:range:clock=20090327T095955Z-
20090327T095955Z;range:clock=20090327T100434Z-
20090327T100434Z;range:clock=20090327T101915Z-20090327T102000Z

The body of the GET_PARAMETER response in the example above specifies three intervals which all occur on the day of 27 March 2009:

- 09:59:55 – 09:59:55 (describes a single point in time)
- 10:04:34 – 10:04:34 (describes a single point in time)
- 10:19:15 – 10:20:00

Two of the returned ranges describe degenerate intervals containing a single time instant which marks a specific point in time. These points in time could, e.g., indicate content positions where there is transient from program to commercial or vice versa. The third and last range describes a closed interval which could, e.g., correspond to the time interval when movie credits are shown.

7. Media Codecs and Formats [Informative]

The BCAST enabler does not specify media codecs and formats.

If BCAST services are distributed over a BDS for which an adaptation specification exists, the adaptation specification MAY include normative text about the use of media codecs and formats.

8. Internet Protocol Usage for File and Stream Distribution Functions

The BSDA MAY use IPv4 and/or IPv6 for stream distribution and file distribution. It is RECOMMENDED that only one version of IP source and destination multicast address be used for the delivery of the same stream distribution session or the same file distribution session.

Terminals SHALL support both IPv4 and IPv6.

9. PUSH Delivery in BCAST

In addition to the file distribution functionality, OMA Push can be used to deliver various types of messages between the network and the terminal via the Interaction Network. Examples include, but are not limited to, 'NotificationMessage' and 'InteractivityMediaDocument' (as defined in [BCAST13-Services]).

The network MAY support OMA Push, and the terminal MAY support it if the terminal supports the Interaction Network. In the case the terminal supports OMA Push, the terminal SHALL support OTA-WSP and MAY support the optional OTA-HTTP method as defined in [OMA Push].

All push messages are delivered to the OMA Push client, which then routes received messages to the BCAST Push client, based on the PUSH Application ID value identifying the BCAST Push client as registered with OMNA:

- URN : x-oma-application:bcast.ua
- code : 0x13

The BCAST Push client is in turn responsible for routing received messages to the correct BCAST terminal functions based on the messages' MIME media types (for example, to the NTC when a message with the MIME media type "application/vnd.oma.bcast.notification+xml" is received).

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-BCAST_Distribution-V1_0-20090212-A	12 Feb 2009	First version approved Ref TP Doc# OMA-TP-2009-0071-INP_BCAST_V1_0_ERP_for_Notification_and_Final_Approval

A.2 Draft/Candidate Version 1.3 History

Note: for the history of the previous releases, please refer to the document of the corresponding version.

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-BCAST-Distribution-V1_3	18Sep2012	all	First baseline for version 1.1 of this spec. This document is based on OMA-TS-BCAST_Distribution-V1_2_0-20120709-D document
	03 Dec 2012	3.3; 4.3; 4.4	Implementation of CR: OMA-BCAST-2012-0064-CR_Scope_BCAST1.3_Distribution.doc
	17 Dec 2013	all	Implementation of CR: OMA-BCAST-2013-0072-CR_CONR1.3_F001_to_F008.doc
Candidate version OMA-TS-BCAST_Distribution-V1_3	14 Jan 2014	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2014-0003-INP_BCAST_V1_3_ERP_and_ETR_for_Candidate_approval

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

B.1 SCR for BCAST File Delivery Client (FD-C)

Note: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (BCAST Distribution System), may override or adapt requirements from this SCR or provide additional requirements.

Item	Function	Reference	Status	Requirement
BCAST-FD-C-001	Support ALC for file distribution over Broadcast Channel	5.2	M	
BCAST-FD-C-002	Support LCT for file distribution over Broadcast Channel	5.2	M	
BCAST-FD-C-003	Support UDP for file distribution over Broadcast Channel	5.2	M	
BCAST-FD-C-005	Support IPv4	8	M	
BCAST-FD-C-006	Support IPv6	8	M	
BCAST-FD-C-007	Support FEC for file distribution over Broadcast Channel	5.2	M	BCAST-FD-C-008 OR BCAST-FD-C-009
BCAST-FD-C-008	Support Compact No-Code FEC Scheme	5.2.2	M	
BCAST-FD-C-009	Support Raptor FEC Scheme	5.2.2	O	
BCAST-FD-C-010	Support FLUTE for file distribution over Broadcast Channel	5.2	M	
BCAST-FD-C-011	Support reception of file metadata in Service Guide	5.2	M	
BCAST-FD-C-012	Support reception of GZIP encoded files	5.2.1	M	
BCAST-FD-C-013	Support Reception Reporting Procedure	5.3.2	O	BCAST-FD-C-015
BCAST-FD-C-014	Support File Repair Procedure	5.3.3	O	BCAST-FD-C-015
BCAST-FD-C-015	Support Associated Delivery Procedure	5.3.1	O	
BCAST-FD-C-016	Support for access to Interaction Channel	5.5	O	BCAST-FD-C-017
BCAST-FD-C-017	Support file distribution over Interaction Channel	5.5	O	
BCAST-FD-C-018	Support FLUTE for file distribution over Interaction Channel	5.5.1	O	

Item	Function	Reference	Status	Requirement
BCAST-FD-C-019	Support HTTP for file distribution over Interaction Channel	5.5.2	O	BCAST-FD-C-020
BCAST-FD-C-020	Support TCP for file distribution over Interaction Channel	5.5.2	O	BCAST-FD-C-021
BCAST-FD-C-021	Support IP for file distribution over Interaction Channel	5.5.2	O	BCAST-FD-C-022 AND BCAST-FD-C-023
BCAST-FD-C-022	Support IPv4	8	O	
BCAST-FD-C-023	Support IPv6	8	O	
BCAST-FD-C-024	Support reception of various types of messages delivered using OMA PUSH	9	O	(BCAST-FD-C-025 AND BCAST-FD-C-026) OR BCAST-FD-C-027
BCAST-FD-C-025	Support Interaction Channel	9	O	
BCAST-FD-C-026	Support OMA PUSH	9	O	MCF : OMA PUSH
BCAST-FD-C-027	Support OTA-HTTP	9	O	
BCAST-FD-C-028	Support File Distribution over Hybrid Broadcast/Interaction Channel	5.6	O	(BCAST-FD-C-001 OR BCAST-FD-C-010) AND BCAST-FD-C-017

B.2 SCR for BCAST File Delivery Application Component in BSA (FDA)

Note: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (BCAST Distribution System), may override or adapt requirements from this SCR or provide additional requirements.

Item	Function	Reference	Status	Requirement
BCAST-FDA-S-001	Support file distribution	5	M	
BCAST-FDA-S-002	Expose back-end interface for file distribution	5.4	O	BCAST-FDA-S-003
BCAST-FDA-S-003	Support back-end interface for file distribution	5.4	O	(BCAST-FDA-S-004 OR BCAST-FDA-S-005) AND BCAST-FDA-S-010
BCAST-FDA-S-004	Support HTTP 1.1 for back-end interface	5.4.1.1	O	BCAST-FDA-S-006
BCAST-FDA-S-005	Support HTTPS for back-end interface	5.4.1.1	O	BCAST-FDA-S-006
BCAST-FDA-S-006	Support TCP for back-end interface	5.4.1.1	O	BCAST-FDA-S-007
BCAST-FDA-S-007	Support IP for back-end interface	5.4.1.1	O	BCAST-FDA-S-008 OR BCAST-FDA-S-009
BCAST-FDA-S-008	Support IPv4	8	O	

Item	Function	Reference	Status	Requirement
BCAST-FDA-S-009	Support IPv6	8	O	
BCAST-FDA-S-010	File distribution back-end interface messages	5.4.1.2	O	

B.3 SCR for BCAST File Distribution Component in BSD/A (FD)

Note: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (BCAST Distribution System), may override or adapt requirements from this SCR or provide additional requirements.

Item	Function	Reference	Status	Requirement
BCAST-FD-S-001	Support file distribution	5	M	
BCAST-FD-S-002	Support ALC for file distribution over Broadcast Channel	5.2	M	
BCAST-FD-S-003	Support LCT for file distribution over Broadcast Channel	5.2	M	
BCAST-FD-S-004	Support UDP for file distribution over Broadcast Channel	8	M	
BCAST-FD-S-005	Support IP for file distribution over Broadcast Channel	8	M	BCAST-FD-S-006 OR BCAST-FD-S-007
BCAST-FD-S-006	Support IPv4	8	O	
BCAST-FD-S-007	Support IPv6	8	O	
BCAST-FD-S-008	Support FEC for file distribution over Broadcast Channel	5.2.2	M	BCAST-FD-S-009 OR BCAST-FD-S-010
BCAST-FD-S-009	Support Compact No-Code FEC Scheme	5.2.2	M	
BCAST-FD-S-010	Support Raptor FEC Scheme	5.2.2	O	
BCAST-FD-S-011	Support FLUTE for file metadata delivery	5.2	O	
BCAST-FD-S-012	Support delivery of file metadata in Service Guide	5.2	O	
BCAST-FD-S-013	Support GZIP content encoding of files	5.2.1	O	
BCAST-FD-S-014	Support Reception Reporting Procedure	5.3.2	O	BCAST-FD-S-016
BCAST-FD-S-015	Support File Repair Procedure	5.3.3	O	BCAST-FD-S-016
BCAST-FD-S-016	Support Associated Delivery Procedure	5.3.1	O	
BCAST-FD-S-017	Expose back-end interface for file distribution	5.4	O	BCAST-FD-S-018

Item	Function	Reference	Status	Requirement
BCAST-FD-S-018	Support back-end interface for file distribution	5.4	O	(BCAST-FD-S-019 OR BCAST-FD-S-020) AND BCAST-FD-S-025
BCAST-FD-S-019	Support HTTP 1.1 for back-end interface	5.4.1.1	O	BCAST-FD-S-021
BCAST-FD-S-020	Support HTTPS for back-end interface	5.4.1.1	O	BCAST-FD-S-021
BCAST-FD-S-021	Support TCP for back-end interface	5.4.1.1	O	BCAST-FD-S-022
BCAST-FD-S-022	Support IP for back-end interface	5.4.1.1	O	BCAST-FD-S-023 OR BCAST-FD-S-024
BCAST-FD-S-023	Support IPv4	8	O	
BCAST-FD-S-024	Support IPv6	8	O	
BCAST-FD-S-025	File distribution back-end interface messages	5.4.1.2	O	
BCAST-FD-S-026	Support file distribution over Interaction Channel	5.5	O	
BCAST-FD-S-027	Support FLUTE for file distribution over Interaction Channel	5.5.1	O	
BCAST-FD-S-028	Support HTTP 1.1 for file distribution over Interaction Channel	5.5.2	O	BCAST-FD-S-029
BCAST-FD-S-029	Support TCP for file distribution over Interaction Channel	5.5.2	O	BCAST-FD-S-030
BCAST-FD-S-030	Support IP for file distribution over Interaction Channel	5.5.2	O	BCAST-FD-S-031 OR BCAST-FD-S-032
BCAST-FD-S-031	Support IPv4	8	O	
BCAST-FD-S-032	Support IPv6	8	O	
BCAST-FD-S-033	Support delivery of deliver various types of messages using OMA PUSH	9	O	BCAST-FD-S-034
BCAST-FD-S-034	Support file distribution over Interaction Channel	9	O	
BCAST-FD-S-035	Support File Distribution over Hybrid Broadcast/Interaction Channel	5.6	O	(BCAST-FD-S-002 OR BCAST-FD-S-011) AND BCAST-FD-S-026

B.4 SCR for BCAST Stream Delivery Client (SD-C)

Note: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (BCAST Distribution System), may override or adapt requirements from this SCR or provide additional requirements.

Item	Function	Reference	Status	Requirement

Item	Function	Reference	Status	Requirement
BCAST-SD-C-001	Support RTP for stream distribution over Broadcast Channel	6.2	M	
BCAST-SD-C-002	Support UDP for stream distribution over Broadcast Channel	6.2	M	
BCAST-SD-C-003	Support IP for stream distribution over Broadcast Channel	6.2	M	
BCAST-SD-C-004	Support IPv4	8	M	
BCAST-SD-C-005	Support IPv6	8	M	
BCAST-SD-C-006	Support RTP payload format for stream distribution over Broadcast Channel	6.2.1	M	BCAST-SD-C-007 OR BCAST-SD-C-008 OR BCAST-SD-C-009
BCAST-SD-C-007	Support codecs used for MBMS	6.2.1	O	
BCAST-SD-C-008	Support codecs used for BCMCS	6.2.1	O	
BCAST-SD-C-009	Support codecs used for IPDC over DVB-H	6.2.1	O	
BCAST-SD-C-010	Support forward error correction	6.2.2	O	
BCAST-SD-C-011	Support buffer control for stream distribution	6.2.3	O	
BCAST-SD-C-012	Support Stream Reception Report	6.3.2	O	(BCAST-SD-C-013 AND BCAST-SD-C-014) OR BCAST-SD-C-015
BCAST-SD-C-013	Support Associated Procedures for Stream Distribution	6.3, 6.3.1	O	
BCAST-SD-C-014	Support HTTP/TCP/IP	6.3.3	O	
BCAST-SD-C-015	Support MBMS Stream Associated Procedures	6.3	O	
BCAST-SD-C-016	Support Interaction Channel	6.5	O	BCAST-SD-C-017
BCAST-SD-C-017	Stream reception over Interaction Channel	6.5	O	
BCAST-SD-C-018	Transparent end-to-end Packet-switched Streaming Service (PSS)	6.5	O	
BCAST-SD-C-019	3G Multimedia Streaming Service	6.5	O	
BCAST-SD-C-020	Support Advisable Time Ranges for Access Switch	6.5.1	O	BCAST-SD-C-017

B.5 SCR for BCAST Stream Delivery Application Component in BSA (SDA)

Note: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (BCAST Distribution System), may override or adapt requirements from this SCR or provide additional requirements.

Item	Function	Reference	Status	Requirement
BCAST-SDA-S-001	Support stream distribution	6	M	
BCAST-SDA-S-002	Expose back-end interface for stream distribution	6.4	O	BCAST-SDA-S-003
BCAST-SDA-S-003	Support back-end interface for non-live stream distribution	6.4.1	O	(BCAST-SDA-S-004 OR BCAST-SDA-S-005) AND BCAST-SDA-S-010
BCAST-SDA-S-004	Support HTTP 1.1 for back-end interface	6.4.1.1	O	BCAST-SDA-S-006
BCAST-SDA-S-005	Support HTTPS for back-end interface	6.4.1.1	O	BCAST-SDA-S-006
BCAST-SDA-S-006	Support TCP for back-end interface	6.4.1.1	O	BCAST-SDA-S-007
BCAST-SDA-S-007	Support IP for back-end interface	6.4.1.1	O	BCAST-SDA-S-008 OR BCAST-SDA-S-009
BCAST-SDA-S-008	Support IPv4	8	O	
BCAST-SDA-S-009	Support IPv6	8	O	
BCAST-SDA-S-010	Stream distribution back-end interface messages	6.4.1.2	O	

B.6 SCR for BCAST Stream Delivery Component in BSD/A (SD)

Note: BCAST adaptation specifications, in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (BCAST Distribution System), may override or adapt requirements from this SCR or provide additional requirements.

Item	Function	Reference	Status	Requirement
BCAST-SD-S-001	Support stream distribution	6	M	
BCAST-SD-S-002	Support RTP for stream distribution over Broadcast Channel	6.2	M	
BCAST-SD-S-003	Support UDP for stream distribution over Broadcast Channel	6.2	M	
BCAST-SD-S-004	Support IP for stream distribution over Broadcast Channel	6.2	M	BCAST-SD-S-005 OR BCAST-SD-S-006
BCAST-SD-S-005	Support IPv4	8	O	

Item	Function	Reference	Status	Requirement
BCAST-SD-S-006	Support IPv6	8	O	
BCAST-SD-S-007	Support RTP payload format for stream distribution over Broadcast Channel	6.2.1	M	BCAST-SD-S-008 OR BCAST-SD-S-009 OR BCAST-SD-S-010
BCAST-SD-S-008	Support codecs used for MBMS	6.2.1	O	
BCAST-SD-S-009	Support codecs used for BCMCS	6.2.1	O	
BCAST-SD-S-010	Support codecs used for IPDC over DVB-H	6.2.1	O	
BCAST-SD-S-011	Support forward error correction	6.2.2	O	
BCAST-SD-S-012	Support buffer control for stream distribution	6.2.3	O	
BCAST-SD-S-013	Support Stream Associated Procedure	6.3	O	(BCAST-SD-S-014 AND BCAST-SD-S-015) OR BCAST-SD-S-016
BCAST-SD-S-014	Support BCAST Stream Associated Procedure	6.3	O	
BCAST-SD-S-015	Support HTTP/TCP/IP	6.3.3	O	
BCAST-SD-S-016	Support MBMS Stream Associated Procedure	6.3	O	
BCAST-SD-S-017	Exposes back-end interface for stream distribution	6.4	O	BCAST-SD-S-018
BCAST-SD-S-018	Support back-end interface for non-live stream distribution	6.4.1	O	(BCAST-SD-S-019 OR BCAST-SD-S-020) AND BCAST-SD-S-025
BCAST-SD-S-019	Support HTTP 1.1 for back-end interface	6.4.1.1	O	BCAST-SD-S-021
BCAST-SD-S-020	Support HTTPS for back-end interface	6.4.1.1	O	BCAST-SD-S-021
BCAST-SD-S-021	Support TCP for back-end interface	6.4.1.1	O	BCAST-SD-S-022
BCAST-SD-S-022	Support IP for back-end interface	6.4.1.1	O	BCAST-SD-S-023 OR BCAST-SD-S-024
BCAST-SD-S-023	Support IPv4	8	O	
BCAST-SD-S-024	Support IPv6	8	O	
BCAST-SD-S-025	Stream distribution back-end interface messages	6.4.1.2	O	
BCAST-SD-S-026	Support Interaction Channel	6.5	O	BCAST-SD-S-027
BCAST-SD-S-027	Stream distribution over Interaction Channel	6.5	O	
BCAST-SD-S-028	Transparent end-to-end Packet-switched Streaming Service (PSS)	6.5	O	
BCAST-SD-S-029	3G Multimedia Streaming Service	6.5	O	

Item	Function	Reference	Status	Requirement
BCAST-SD-S-030	Support Advisable Time Ranges for Access Switch	6.5.1	O	BCAST-SD-S-027

Appendix C. MIME Media Types

C.1 Media-Type Registration Request for application/vnd.oma.bcast.associated-procedure-parameter+xml

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name: application

Subtype name: vnd.oma.bcast.associated-procedure-parameter+xml

Required parameters: none

Optional parameters: none

Encoding considerations: binary

Security considerations:

OMA BCAST Associated Delivery Procedure Parameters are passive, meaning they do not contain executable or active content which may represent a security threat. The format does not include confidential fields. However, the information present in this media format is used to configure the receiving application. Thus, the usage of the format is vulnerable to attacks modifying or spoofing the content of this format. Depending on the system architecture, it is recommended to use source authentication and integrity protection.

Interoperability considerations:

This content type carries associated delivery procedure parameters within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements and interoperability test cases for this content.

Published specification:

OMA BCAST 1.0 Enabler Specification – File and Stream Distribution for Mobile Broadcast Services, especially section 5.3. Available from http://www.openmobilealliance.org

Applications, which use this media type:

OMA BCAST File Distribution Services

Additional information:

Magic number(s): none

File extension(s): none

Macintosh File Type Code(s): none

Intended usage: Limited use.

Only for usage with OMA BCAST File Distribution Services , which meet the semantics given in the mentioned specification.

Person & email address to contact for further information:

Uwe Rauschenbach, uwe.rauschenbach@nsn.com

Author/Change controller:

OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

C.2 Media-Type Registration Request for application/vnd.oma.bcast.simple-symbol-container

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name:	application
Subtype name:	application/vnd.oma.bcast.simple-symbol-container
Required parameters:	none
Optional parameters:	none
Encoding considerations:	binary

Security considerations:

OMA BCAST File Repair Response Messages shall be passive, meaning they do not contain executable or active content which may represent a security threat. The content of this media type are either source or repair symbols part of a binary file object. Thus, its security considerations depend on the security requirements on the file object and on the actual system architecture. 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 erroneous repair symbols.

Interoperability considerations:

This content type carries file repair response messages within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements and interoperability test cases for this content.

Published specification:

OMA BCAST 1.0 Enabler Specification – File and Stream Distribution for Mobile Broadcast Services, especially section 5.3. Available from http://www.openmobilealliance.org

Applications, which use this media type:

OMA BCAST File Distribution Services

Additional information:

Magic number(s): none

File extension(s): none

Macintosh File Type Code(s): none

Intended usage: Limited use.

Only for usage with OMA BCAST File Distribution Services , which meet the semantics given in the mentioned specification.

Person & email address to contact for further information:

Uwe Rauschenbach, uwe.rauschenbach@nsn.com

Author/Change controller:

OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

Appendix D. Distribution in Hybrid Broadcast/Interactive Scenarios (INFORMATIVE)

D.1 File Distribution in Hybrid Broadcast/Interactive Scenario

In a hybrid Mobile TV system it is assumed that a set of file distribution services is available over both broadcast channel and interaction channel. See section Appendix I of [BCAST13-SG] for a more extensive description of a hybrid Mobile TV system and its typical characteristics.

Over the broadcast channel, a file download service is delivered using either FLUTE or ALC as the transport protocol (see chapter 5). Over the interaction channel a file download service is delivered using either FLUTE (see section 5.5.1) or HTTP (see section 5.5.2). By providing a file distribution service over the broadcast channel as well as the interaction channel, the BCAST server is able to realize hybrid broadcast/unicast scenarios where the interaction channel, e.g., is used as a fallback delivery method in case the terminal goes out of broadcast channel coverage. Section I.3.3 of [BCAST13-SG] contains an example that describes how the Service Guide is instantiated in order to support such a scenario.

For a hybrid broadcast/unicast file distribution service, there are some characteristics of FLUTE and HTTP to consider before choosing the delivery method for interaction access. These characteristics are to be considered independently of the choice of delivery method for the broadcast channel.

- Using FLUTE for file distribution over interaction channel leads to the terminal not being able to control which data that is sent from the server. Even if the server synchronizes the file delivery over broadcast channel with the file delivery using FLUTE over interaction channel, there is still a possibility that the terminal fails to receive file fragments during the transition between broadcast access and interaction access. The terminal is then unable to specifically retrieve these missing file fragments.
- Using HTTP for file distribution over interaction channel is identical to the file repair procedure, as specified in section 5.5. This means that the terminal is able to retrieve complete files as well as individual file fragments depending on its need. Although, unlike FLUTE there is no possibility to receive updated file description in-band of the HTTP session. The file descriptions needs to be signaled in the Service Guide instead and the terminal needs to receive an updated Service Guide before changes can be detected.

Considering the characteristics mentioned in the bullets above, FLUTE as the transport protocol for file distribution over the interaction channel in a hybrid broadcast/unicast scenario is likely to be more suitable for file distribution cases with a changing set of files and versions continuously delivered in a carousel, while HTTP is likely to be more suitable for fixed content distribution cases where the set of files and their version do not change.

D.2 Stream Distribution in Hybrid Broadcast/Interactive Scenario

In a hybrid Mobile TV system it is assumed that a set of stream distribution services is available over both broadcast channel and interaction channel. See Appendix I of [BCAST13-SG] for a more extensive description of a hybrid Mobile TV system and its typical characteristics.

This section describes some methods to improve switching time as well as methods for improving the user experience when switching service access in such a hybrid Mobile TV system.

D.2.1 Initial BDS Selection

If a particular service is available over both broadcast channel and interaction channel, it is up to the terminal to select either of them for delivery of the service. The choice may be based on several technical or business related properties of the bearers (e.g. bit rate, monetary cost), or it may be based on user decision. For example, the interactive channel might offer a streaming service at a different bit rate and at a different monetary cost. A terminal is likely to choose broadcast access unless interaction access has been determined beneficial from a user experience point of view. The broadcast coverage may for example suffer from low quality and the terminal therefore chooses the interaction access to avoid the ping-pong effect

caused by continuous access switches, which in turn are caused by broadcast access being lost and reacquired over and over again.

The terminal may setup a stream distribution session delivered over the interaction channel at the start of a Mobile TV session independent of the initial choice of access. This is done in order to decrease the time for switching from broadcast access to interaction access as described in section D.2.2. Otherwise the terminal will have to setup the interaction channel session at the time of loosing broadcast coverage, or when detecting that there is a high probability that the broadcast coverage will be lost.

D.2.2 Switching from Broadcast Access to Interaction Access

The access change from broadcast channel to interaction channel typically occurs suddenly and without warning when the available signal strength is no longer sufficient for broadcast access. The terminal may for example identify the loss of broadcast access by measuring signal strength and/or packet loss and compare the measurements against a predefined threshold, or by noticing that RTP reception has stopped completely. As a result the terminal may use the Service Guide to identify that the stream distribution flow is alternatively available over the interaction channel (see section I.3.2 in [BCAST13-SG]), terminate the broadcast session, and initiate reception over the interaction channel instead. It might be up to the user to decide whether the switch of access is desired or if the session shall be terminated, especially if delivery over the interaction channel induces additional monetary cost.

During the change of access to interaction channel it may be that a drop-out of the service will last corresponding to the time it takes the terminal to setup reception over interaction channel, possibly change codecs, and to fill the receiver buffers to a suitable level. This could be in the order of several seconds. As previously mentioned, one way for the terminal to decrease this time period is to setup a stream distribution session delivered over the interaction channel at the start of a Mobile TV session independent of the initial choice of access. In case the terminal is able to access the requested service over the broadcast channel, the interaction channel session is left in a paused state. The only signaling between the terminal and the server is then “keep alive” messages to prevent the server from timing out. If broadcast coverage is lost, the Mobile TV application continues the service reception over the interaction channel. This is done by merely issuing RTSP PLAY to the streaming server since the required number of SETUP methods has already been exchanged between the client and server. It may happen that the user have switched service since the initial setup leading to the unicast session that was originally setup no longer refers to the broadcast session being received. Using the Fast Content Switching mechanism described in section 6.5 the terminal then issues a PLAY with the URL for a unicast session that corresponds to the lost broadcast session. When broadcast coverage later is reacquired the terminal may send RTSP PAUSE to stop unicast reception and once again receive the service over broadcast access.

The BCAST server may provide functionality for time-shifting of live streaming sessions as described in section 6.5. When both the BCAST server and the terminal support Range headers expressed in UTC it is possible for the terminal to resume the session over interaction channel at the time instant where the broadcast access was lost. Thus there may be a blackout during the access change, but there is no loss of media

D.2.3 Switching from Interaction Access to Broadcast Access

If the terminal receives a service over interaction channel, and it recognizes that broadcast channel coverage is reacquired, it may terminate or pause the interaction channel session and initiate reception over the broadcast channel. The decision to switch, and when to switch, is made at the terminal side. It may be based on several technical or business related properties of the bearers (e.g. bit rate, monetary cost), or it may be based on user decision.

An access change from interaction channel to broadcast channel has the possibility to be made nearly seamless when both broadcast and unicast streaming can be received by the client at the same time. Although, the terminal might need to change codecs or codec settings just as for the switch from broadcast access to interaction access.

As specified in section 6.5.1, a BCAST server may signal positions in the content where a controlled switch to broadcast access would have the least impact on the user experience. Those signalled instants in time mark positions at which a switch would have a low impact on the viewing experience and are typically mapped to transitions between programs and commercials and time intervals when movie credits are shown. By retrieving this information and using it to plan the controlled access switch from interaction access to broadcast access, the terminal avoids performing the switch during vital parts of the content such as interesting and important scenes of a movie. How these instants in time are determined at the network side is out of scope of this specification, but content providers are one possible source of information.

D.2.4 Synchronization of Stream Distribution Flows

One of the challenges of making an access change of a service, being either a change from broadcast channel to interaction channel or vice versa, is to align the two media flows at the point of change in order to guarantee a good user experience. One method for aligning the flows at the RTP level is described in the following paragraphs.

For each media component in the stream distribution flow from which the switch is made, the terminal compares the SSRC value against the corresponding media component of the stream distribution flow to which the switch is made. Two different outcomes are possible:

1) The terminal detects two different SSRC values. The terminal assumes that the same wall clock time is used for the two media components and waits for media component specific RTCP packets to be received in order to align the media components using the wall clock time.
2) The terminal detects that the same SSRC value is used. The terminal assumes that the same random RTP timestamp offset is used for the media components and align the media components using the RTP timestamp. This case enables the possibility for the terminal to make use of media packets that are located in the media buffer at the time of switching access, i.e., media packets that were received from the media flow from which the switch was made.

The most straight forward way for the terminal to align two media components, with regards to codec handling, would be if they are encoded using the same codec and codec settings. In addition, if the SSRC of the two flows are the same, as described in 2) above, it would further help the terminal to perform an access switch with a good user experience.