



File Distribution and Stream Distribution

Draft Version 1.0 – 07 NOV 2005

Open Mobile Alliance
OMA-TS-BCAST_Distribution-V1_0-20051107-D

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2004 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

1. SCOPE	5
2. REFERENCES	6
2.1 NORMATIVE REFERENCES	6
2.2 INFORMATIVE REFERENCES	7
3. TERMINOLOGY AND CONVENTIONS	8
3.1 CONVENTIONS	8
3.2 DEFINITIONS	8
3.3 ABBREVIATIONS	8
4. INTRODUCTION	9
5. FILE DISTRIBUTION	10
5.1 FUNCTIONAL ARCHITECTURE OF FILE DISTRIBUTION	10
5.1.1 General	10
5.2 PROTOCOL FOR FILE DISTRIBUTION	10
5.2.1 Introduction	10
5.2.2 Detailed description about the usage of Protocol	11
5.3 PROCEDURES FOR FILE DISTRIBUTION	11
5.3.1 Procedure 1	Error! Bookmark not defined.
5.3.2 Procedure 2	Error! Bookmark not defined.
5.3.3 Procedure	Error! Bookmark not defined.
6. STREAM DISTRIBUTION	26
6.1 FUNCTIONAL ARCHITECTURE OF STREAM DISTRIBUTION	26
6.1.1 General	26
6.2 PROTOCOL FOR STREAM DISTRIBUTION	26
6.2.1 Introduction	26
6.2.2 Detailed description about the usage of Protocol	Error! Bookmark not defined.
6.3 PROCEDURES FOR STREAM DISTRIBUTION	27
6.3.1 Procedure 1	27
6.3.2 Procedure 2	27
6.3.3 Procedure	27
7. MEDIA CODECS AND FORMATS [INFORMATIVE]	28
7.1 GENERAL	28
7.2 SPEECH	28
7.3 AUDIO	28
7.4 VIDEO	28
7.5 STILL IMAGES	28
7.6 TEXT	28
7.7 OTHER MEDIA CODECS AND FORTMATS	28
8. SUPPLEMENTATY METHODS AND PROCEDURES FOR FILE DISTRIBUTION AND STREAM DISTRIBUTION	29
8.1 GENERAL	29
8.2 ERROR RESILIENCE	29
8.2.1 Methods for Error resilience	29
8.3 ERROR REPORTING	29
8.4 OTHERS	29
APPENDIX A. CHANGE HISTORY (INFORMATIVE)	30
A.1 APPROVED VERSION HISTORY	30
A.2 DRAFT/CANDIDATE VERSION <CURRENT VERSION> HISTORY	30
APPENDIX B. STATIC CONFORMANCE REQUIREMENTS (NORMATIVE)	31
B.1 SCR FOR XYZ CLIENT	31
B.2 SCR FOR XYZ SERVER	31

APPENDIX C. <ADDITIONAL INFORMATION>.....32

 C.1 APP HEADERS.....32

 C.1.1 More Headers.....33

Figures

Figure 1: Example Figure Error! Bookmark not defined.

Tables

Table 1: Example Table Error! Bookmark not defined.

1. Scope

~~<< Define as it relates to Open Mobile Alliance Activity. If it adds clarity, define what is not in the scope. DELETE THIS COMMENT >>~~

The scope of this specification is the Stream and File Distribution functionality of OMA Mobile Broadcast (BCAST) Enabler. Referring to OMA BCAST Architecture, this document normatively specifies the interfaces FD-1, FD-2, FD-4, FD-5, FD-6, SD-1, SD-2, SD-5 and SD-6 [OMA BCAST AD].

Media codecs and related media-specific transport payloads are out of scope of this specification. ~~Also, the use of interaction channel to realize the file and stream distribution of OMA BCAST Services is out of scope.~~

2. References

The policy for reference lists is:

1. OMA documents listed should have at least one approved version - draft-only docs should not be referenced. Exception exists for documents that will be approved with or after the referenced doc is approved (may be part of same enabler package). In short - approved docs should not reference unapproved docs.
2. When a reference is made to an OMA specification, then Open Mobile Alliance with the TM symbol (™) should be used in the description.
3. The name + version (no date) for OMA specifications are generally sufficient - dates should be used only if there is a specific reason to limit the usage.
4. For references to WAP Forum docs, dates should not be included as DID's for the old WAP Forum specifications are enough and the reference description should refer to WAP Forum™.
5. References to other affiliate docs should similarly provide sufficient information to uniquely determine the needed document and should provide the appropriate source information.
6. The URL for OMA material (new OMA and affiliate) should always be <http://www.openmobilealliance.org> (an exception is OMNA that is reached through <http://www.openmobilealliance.org/tech/omna>)

Models to use

[REFLABEL] <General Model> "Ref Title", Ref information (source, date, id),
URL: http://<ref-source>/

[OMADOC] <OMA Model> "OMA Document Title", Open Mobile Alliance™,
OMA-<docname>{-<version>}, URL:<http://www.openmobilealliance.org/>

If there are no entries in the table - enter 'none' to be clear.

DELETE THIS COMMENT

2.1 Normative References

- | | |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [IOPPROC] | “OMA Interoperability Policy and Process”, Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL:http://www.openmobilealliance.org/ |
| [RFC2119] | “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt |
| [RFC2234] | “Augmented BNF for Syntax Specifications: ABNF”. D. Crocker, Ed., P. Overell. November 1997, URL:http://www.ietf.org/rfc/rfc2234.txt |
| [RFC 3450] | IETF RFC 3450, “Asynchronous Layered Coding (ALC) Protocol Instantiation” |
| [RFC 3451] | IETF RFC 3451, “Layered Coding Transport (LCT) Building Block” |
| [RFC 3452] | IETF RFC 3452, “Forward Error Correction (FEC) Building Block” |
| [RFC 3926] | IETF RFC 3926, “FLUTE - File Delivery over Unidirectional Transport” |

[RFC 3550] IETF RFC 3550, “RTP: A Transport Protocol for Real-Time Applications”
[3GPP MBMS] 3GPP TS 26.346 v6.1.0, “MBMS Protocols and Codecs”
<< Add/Remove reference rows as needed! >>

2.2 Informative References

[\[OMA BCAST AD\]](#) [“Mobile Broadcast Services Architecture”, Version 1.0, Open Mobile Alliance™](#)
<< Add/Remove reference rows as needed! >>

3. Terminology and Conventions

3.1 Conventions

~~<< If doc includes normative material keep the next two paragraphs. DELETE THIS COMMENT >>~~

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 [RFC2119].

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

~~<< OR if doc is informative just keep the next line. DELETE THIS COMMENT >>~~

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

~~<< If needed, describe or declare using appropriate normative references the additional conventions that are used. DELETE THIS COMMENT >>~~

3.2 Definitions

~~<< Add definitions in new rows of the following table as needed. DELETE THIS COMMENT >>~~

Term 1	Definition
Term 2	Definition

3.3 Abbreviations

~~<< Add abbreviations as needed to the following table. DELETE THIS COMMENT >>~~

OMA	Open Mobile Alliance
xxx FD	xxx File Delivery Function (Network side)
FD-C	File Delivery Client Function (Terminal side)
SD	Stream Distribution Function (Network side)
SD-C	Stream Distribution Client Function (Terminal side)

4. Introduction

~~← From a market perspective...~~

~~oWhat can you do with this specification?~~

~~oWhat problem does this solve?~~

~~oHow can this specification be applied?~~

~~oConsider the target audience and provide deployment examples as possible.~~

~~DELETE THIS COMMENT→~~

This specification defines the Stream and File Distribution function for OMA Mobile Broadcast Services. Stream distribution is a functionality that enables the delivery of real-time A/V streams from system to terminals. File distribution is similar functionality to stream distribution with the difference that instead of real-time media streams discrete media objects and files are to be 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 interactive channel to deliver the services is not specified in this specification. However,~~ The use of interactive channel to realize the associated procedures for stream and file distribution is specified in this document

This specification is agnostic to any underlying Broadcast Distribution System – it only assumes the capability of delivering IP streams from server to terminal(s). That way this specification provides common way to achieving the delivery of files and distribution of streams over various different systems and therefore consequently enables a greater level of interoperability.

5. File Distribution

~~<<Sections for the normative specification text. Fill in as needed. The following validates the styles used for the headers. DELETE THIS COMMENT>>~~

~~5.1 Functional Architecture of File Distribution~~

~~5.1.1 General~~

~~<text>~~

~~<text>~~

~~5.1.1.1 Example Level 4~~

~~<text>~~

5.2 Protocol for File Distribution

5.2.15.1

Int

roduction

The specification for OMA BCAST File Distribution function consists of specification for five interfaces, FD-1, FD-2, FD-4, 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 Distribution Client (FD-C) on the Terminal side and File Distribution ~~Component Server (FD-SFD)~~ on the Network side. These interfaces are specified in sections 5.2 and 5.3. The interfaces FD-1, FD-2 and FD-4 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 Server (FD-SFD)~~ and the File Application (FA), both on the Network side. These back-end interfaces are specified in section 5.4.

5.2 File Distribution over Terminal-Network interfaces

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

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 that BCAST File Distribution is provided, and thus congestion control is not used for 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 no-code FEC code that simply sends the original data using no FEC coding. In addition to FEC protection over broadcast channel, 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 ~~Terminal~~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 File Delivery Table, and, the file delivery session is a FLUTE session as specified in [3GPP MBMS].
- The ~~Terminal~~FD-C SHALL support the delivery of metadata associated with files in the Service Guide as specified in the OMA BCAST Service Guide [Note: reference to relevant section in BCAST TS Service Guide to be added.]

The ~~File Delivery function in BSDA~~FD-SFD SHALL support either one or both of the above-mentioned methods for signalling the file parameters.

[Note: DVB is also specifying the content delivery and service guide delivery that includes ALC. Should there be any variation in the usage of ALC in DVB vs. 3GPP, it shall be later resolved in this section.]

5.2.2 Detailed description about the usage of Protocol

<text>

5.2.2.1 Example Level 4

<text>

5.3 Procedures for File Distribution

Note: It remains to be discussed whether it is mandatory or optional to implement file repair and reception reporting in terminals and/or systems.

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, and in association with, BCAST delivery methods and their sessions. Those procedures that shall 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~~ **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.

~~BCAST Terminals~~ **FD-C** ~~shall~~ **SHALL** support the download file repair procedure as defined in section 5.3.2

~~BCAST Terminals shall~~ **FD-C** **SHALL** support the download reception reporting procedure as defined in section 5.3.3

5.3.1 ~~5.3.1~~ Associated Procedure Description

An associated procedure description instance (configuration information) for the 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, 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 ~~terminal~~ **FD-C** cross-referencing of in and out-of-band configuration files.

The MIME application type "application/BCAST-associated-procedure-parameter" 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 section 5.3.4.

~~5.3.2~~ ~~5.3.2~~ Reception reporting

Following successful reception of content, a reception reporting procedure ~~may~~ **MAY** be initiated by the ~~BCAST Terminal~~ **FD-C** to the ~~BSDA~~ **FD-SFD**.

For BCAST Download Delivery method, the reception reporting procedure is used to report the complete reception of one or more files.

If the ~~BSDA~~ **FD-SFD** provided parameters requiring reception reporting confirmation then the ~~BCAST Terminal shall~~ **FD-C** **SHALL** confirm the content reception.

If reception reporting is requested for statistical purposes the ~~BSDA~~ **FD-SFD** may specify the percentage subset of ~~BCAST Terminals~~ **FD-Cs** it would like to perform reception reporting.

Transport errors can prevent a ~~BCAST Terminal~~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. A ~~BCAST Terminal~~FD-C ~~shall~~ **SHALL** behave according to the information it has even when it is aware that this may be incomplete.

The ~~BCAST Terminal~~FD-C:

1. Identifies the complete reception of a content item (e.g. a file). See sections 5.3.2.1 and 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 a ~~server~~FD-SFD 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 ~~server~~FD-SFD at the selected time. See section 5.3.2.6

Then the ~~server~~FD-SFD:

1. Responds with a *reception report response* message either containing the requested data, or alternatively, describing an error case. See section 5.3.2.7

5.3.2.1-5.3.2.1

Ide

ntifying Complete File Reception from BCAST Download

~~FD-C SHALL A file is~~ determined file to be 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 ~~terminal~~FD-C to compile the RACK reception report for that file.

5.3.2.2-5.3.2.2

Ide

ntifying Complete BCAST Delivery Session Reception

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

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

5.3.2.3-5.3.2.3

Det

etermining Whether a Reception Report Is Required

Upon full reception of a content item or when a session is complete, the ~~BCAST Terminal~~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).

A delivery method may associate zero or one associated delivery procedure descriptions with a BCAST delivery session. Where an associated delivery procedure description is associated with a session, and the description includes a *postReceptionReport* element, the ~~terminal shall~~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 ~~terminal~~FC-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. StaR-all is relevant to both streaming and download delivery.

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 [Terminals-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 [terminal-FD-C](#) which entered the associated session [shall SHALL](#) send a reception report. If the *samplePercentage* were provided for reportType StaR and StaR-all and the value is less than 100, the [terminal-FD-C](#) generates a random number which is uniformly distributed in the range of 0 to 100. The [terminal-FD-C](#) sends the reception report when the generated random number is of a lower value than the *samplePercentage* value.

[5.3.2.4.5.3.2.4](#)

Req

uest Time Selection

The [BCAST-Terminal-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 5.3.3.4. The *offsetTime* and *randomTimePeriod* used for delivery confirmation may have different values from those used for file-repair and are signalled separately in the reception reporting description of the associated delivery procedure description instance.

In general, reception reporting procedures may be less time critical than file repair procedures. The default behaviour is that a [terminal-shall-FD-C SHALL](#) stop its postReceptionReport timers which are active when a postFileRepair timer expires, which results in the successful initiation of point-to-point communications between [terminal-FD-C](#) and [BSDA-FD-SFD](#).

In some circumstances, the system bottleneck may be in the [server-FD-SFD](#) 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 terminal-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.

[5.3.2.5.5.3.2.5](#)

Rec

ption Report Server Selection

Reception report server selection is performed according to the procedure described in section 5.3.3.5.

5.3.2.6

Rec

Reception Report Message

Once the need for reception reporting has been established, the ~~FD-C SHALL BCAST Terminal~~ sends one or more Reception Report messages to the ~~BSDA~~~~FD-SFD~~. 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. URI is required to uniquely identify the file (resource).

The ~~Terminal shall~~~~FD-C SHALL~~ make a Reception Report request using the HTTP (RFC 2616) POST request carrying XML formatted metadata for each reported received content (file). An HTTP session shall be used to confirm the successful delivery of a single file. If more than one file were downloaded in a particular BCAST download multiple descriptions shall be added in a single POST request.

Each Reception Report is formatted in XML according the following XML schema (section 5.3.4).

Multipart MIME (multipart/mixed) may be used to aggregate several small XML files of reception reports to a larger object.

For Reception Acknowledgement (RAck) a receptionAcknowledgement element shall provide the relevant data.

For Statistical Reporting (StaR) 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):

- 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 = "download" || "streaming" || "mixed".
- The *serviceId* attribute is value and format is taken from the respective userServiceDescription serviceID definition.
- The *clientId* attribute is unique identifier for the ~~Terminal~~~~FD-C~~.
- The *serverURI* attribute value and format is taken from the respective associatedDeliveryProcedureDescription serverURI which was selected by the ~~terminal~~~~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

Fil

e repair

Editorial note: 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 for the case when file information is delivered in the service guide rather than in the FLUTE FDT table. This is TBD.

5.3.3.1 ~~5.3.3.1~~ Introduction

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

- Feedback implosion due to a large number of BCAST TerminalsFD-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 TerminalsFD-C requests.
- File repair server overload, caused again by the incoming and outgoing traffic due to the Terminals'FD-Cs' requests arriving at the serverFD-SFD, and the serverFD-SFD 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 serversFD-SFDs.

The BCAST TerminalFD-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 BCAST TerminalFD-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 TerminalFD-C to an BCAST download session, redirecting the TerminalFD-C to another serverFD-SFD, or alternatively, describing an error case.

The BSDAFD-SFD 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 ~~5.3.3.2~~ Identification of End of Transmission for BCAST Download Delivery

FLUTE File Delivery Table (FDT) Instances include an "expires" attribute, which defines the expiration time of the FDT instance. The sender must use an expiry time relative to its sender current time. The Sender Current Time header field shall be present in all FLUTE packets containing data of an FDT Instance. According to RFC 3926, "the receiver SHOULD NOT use a received FDT Instance to interpret packets received beyond the expiration time of the FDT Instance".

The ~~BCAST-terminal~~FD-C determines the end-of-transmission for the BCAST download delivery based on the expiration time of the FDT instance and any end-of-object (B-flag) and end-of-session (A-flag, and SDP end time) information available, or based on file delivery parameters conveyed in the Service Guide.

Note: this is yet to be specified in the SG

When a particular file (URI) is present in several FDT Instances with different TOI values, then the expiration time of the FDT Instances, which contain the highest TOI value of that file, determines the end of transmission time for that file. A ~~terminal~~FD-C ~~shall~~SHALL only determine transmission completeness for a file for the most up-to-date instance of the file (the file instance/version with the highest/most-up-to-date TOI) - and shall not use FDT Instance expiry time to determine transmission completeness for any other (TOI) instances of a file (fileURI). The intention of this is to just start the Associated Delivery Procedure back-off timer for the latest version of a file.

When a particular file (URI) is present in more than one FDT Instance with the same TOI value, then the end of transmission time is defined by the expiration time of the last FDT Instance to expire.

If an FDT Instance is received describing the file after this time (giving an FDT Instance expiry time in the future and the same or a higher TOI value) the ~~terminal~~FD-C ~~shall~~SHALL determine that the transmission of the file is incomplete - i.e. that more packets may arrive by the BCAST download session for that file, 'forgetting' its previous file transmission complete determination. This effectively resets and stops any running timers already initiated for an associated delivery procedure for that file.

If the ~~BCAST-terminal~~FD-C receives an end-of-object packet (with FLUTE header B flag set true) the ~~BCAST-terminal~~~~shall~~FD-C SHALL determine that the transmission of that object is complete, and shall interpret that as file transmission complete if no, more recent, TOIs are described for the same file (URI) in any received and unexpired FDT Instance(s).

If the ~~FD-C~~BCAST ~~terminal~~ determines that the download session is complete (as specified in section 5.3.2.2) then it shall interpret this also that all the transmissions of all files (and TOIs) described by all FDT Instances, received from that session, are complete.

5.3.3.3 ~~5.3.3.3~~ Identification of Missing Data from a BCAST Download

The session description and the BCAST download delivery protocol provide the ~~Terminal~~FD-C with sufficient information to determine the source block and encoding symbol structure of each file. From this a ~~Terminal~~FD-C is able to determine which source symbols should have been transmitted but have not been received. The ~~Terminal~~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 ~~BCAST-Terminal~~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).

5.3.3.4 ~~5.3.3.4~~ Back-off Timing the Procedure Initiation Messaging for Scalability

This section describes a *back-off mode* for BCAST download to provide information on when a ~~Terminal~~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 how the information and method a ~~BCAST-Terminal~~FD-C uses to calculate a time (*back-off time*), instance of the back-off mode, to send a file repair message to the ~~BCAST-server~~FD-SFD.

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 ~~BCAST-Terminal~~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 ~~BCAST TerminalFD-C shall~~ SHALL generate random or pseudo-random time dispersion of *repair requests* to be sent from the ~~TerminalFD-C (BCAST Terminal)~~ to the ~~sender (BCAST server)~~ FD-SFD. 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 ~~TerminalFD-C~~) enhances system scalability to the total number of such messages the system can handle without failure.

5.3.3.4.1 ~~5.3.3.4.1~~ Offset time

The *OffsetTime* refers to the repair request suppression time to wait before requesting repair, or in other words, it is the time that a ~~BCAST TerminalFD-C shall~~ 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.

5.3.3.4.2 ~~5.3.3.4.2~~ Random Time Period

The *Random Time Period* refers to the time window length over which a ~~BCAST TerminalFD-C shall~~ 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 ~~BCAST TerminalFD-C shall~~ SHALL calculate a uniformly distributed *Random Time* out of the interval between 0 and *Random Time Period*.

5.3.3.4.3 ~~5.3.3.4.3~~ Back-off Time

The sending of the file *repair request* message shall start at $Back-off\ Time = offset-time + Random\ Time$, and this calculated time shall be a relative time after the BCAST data transmission. The ~~BCAST TerminalFD-C shall not~~ SHALL NOT start sending the repair request message before this calculated time has elapsed after the initial transmission ends.

5.3.3.4.4 ~~5.3.3.4.4~~ Reset of the Back-off Timer

The reception of an updated (higher version number) associated *DeliveryProcedureDescription* 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.

5.3.3.5 ~~5.3.3.5~~ File Repair Server Selection

5.3.3.5.1 ~~5.3.3.5.1~~ List of Server URIs

A list of file repair servers is provided by a list of server URIs as attributes of the Associated Delivery procedure description. These attributes and elements specify URIs of the file repair servers. Server URIs 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 "serverURI" elements, each of which shall be a child-element of the "procedure" element. The "serverURI" element provides the references to the file repair server via the "xs:anyURI" value. At least one "serverURI" element shall be present.

5.3.3.5.2 ~~5.3.3.5.2~~ Selection from the Server URI List

The ~~BCAST-TerminalFD-C~~ randomly selects one of the server URIs from the list, with uniform distribution.

5.3.3.6 ~~5.3.3.6~~ File Repair Request Message

Once missing file data is identified, the ~~BCAST-TerminalFD-C~~ sends one or more messages to a file repair server requesting transmission of data that allows recovery of missing file data. All file repair requests and repair responses for a particular 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 "serverURI".

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

When a ~~BCAST-TerminalFD-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 ~~BSDA-FD-SFD~~ may use these and/or redundant symbols in providing the necessary repair data.

5.3.3.6.1 ~~5.3.3.6.1~~ File Repair Request Message Format

After the BCAST download session, the ~~TerminalFD-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 the download delivery method (the FLUTE FDT Instances describe file URIs). The (SBN, ESI) pair uniquely identifies an encoding symbol. For completely missed files, a Repair Request may give only the URI of the file.

The ~~TerminalFD-C~~ makes a file repair request using the HTTP (RFC 2616) request method GET. If specific symbols are requested, the (SBN, ESI) of requested encoding symbols are URL-encoded (RFC 1738) 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 FLUTE session a file with URI = www.example.com/news/latest.txt was delivered to an ~~BCAST-TerminalFD-C~~. After the FLUTE session, the ~~BCAST-TerminalFD-C~~ recognized that it did not receive two packets with SBN = 5, ESI = 12 and SBN=20, ESI = 27. Then the HTTP GET request is as follows:

GET www.example.com/news/latest.txt?bcast-File-repair&SBN=5;ESI=12&SBN=20;ESI=27
HTTP/1.1

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 were downloaded in a particular BCAST download session, and, if the ~~BCAST-TerminalFD-C~~ needs repair data for more than one file received in that session, the ~~BCAST-TerminalFD-C~~ shall send separate HTTP GET requests for each file.

An HTTP ~~Terminal~~-implementation with 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 ~~BCAST-TerminalFD-C~~ shall SHALL distribute the URL-encoded data into multiple HTTP GET requests.

In any case, all the HTTP GETs of a single file repair session shall be performed within a single TCP session and they shall be performed immediately one after the other.

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

In this case an HTTP GET with a normal query shall be used to request the missing data.

The general HTTP URI syntax is as follows RFC 2616 :

- `http_URL = "http:" "/" host [":" port] [abs_path ["?" query]]`

Where, for BCAST File Repair Request:

- `query = application *("&" sbn_info)`
- `application = "bcast-file-repair"`
- `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

Thus, the following symbols adopt a special meaning for BCAST: ? - + , ; & =

One example of a query on encoding symbol 34 of source block 12 of a music file "number1.aac" is:

- `http://www.operator.com/greatmusic/number1.aac?bcast-file-repair&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 (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).

5.3.3.7 ~~5.3.3.7~~ File Repair Response Message

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

An ~~BCAST Terminal~~FD-C ~~shall~~ SHALL be prepared for any of these 4 response scenarios:

- The ~~server FD-SFD~~ returns a repair response message where a set of encoding symbols forms an HTTP payload as specified below.
- The ~~server FD-SFD~~ redirects the Terminal to a broadcast/multicast delivery (an BCAST download session).
- The ~~server FD-SFD~~ 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 TBD describes the case of no ~~server FD-SFD~~ response).

For (reasonably) uniformly distributed random data losses, immediate point-to-point HTTP delivery of the repair data will generally be suitable for all ~~Terminal FD-Cs~~. 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 ~~BSDA FD-SFD~~ prefers to handle repairs after that repeat BCAST download.
- Many ~~terminal FD-Cs~~ 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 ~~terminal FD-Cs~~ that have made a repair request would be advantageous.

5.3.3.7.1 ~~5.3.3.7.1~~ 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 ~~Terminal 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 ~~terminal 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.

5.3.3.7.2 ~~5.3.3.7.2~~ File Repair Response Message Format for HTTP Carriage 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 shall be set to "application/simpleSymbolContainer", which denotes that the message body is a simple container of encoding symbols as described below.

This header is as follows:

- HTTP/1.1 200 OK
- Content-Type: application/simpleSymbolContainer
- 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 [terminalFD-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 AAA illustrates the complete file repair response message format (box sizes are not indicative of the relative lengths of the labelled entities).

HTTP Header		
Length Indicator	FEC Payload ID	Encoding Symbols
Length Indicator	FEC Payload ID	Encoding Symbols
Length Indicator	FEC Payload ID	Encoding Symbols

- Length Indicator** (2 bytes): indicates the number of encoding symbols in the group (in network byte order, i.e. high order byte first)
- FEC Payload ID**: indicates which encoding symbols are included in the group. The format and interpretation of the FEC Payload ID are dependent on the FEC Scheme in use.
- Encoding Symbols**: contain the encoding symbols. All the symbols shall be the same length.

Figure AAA: File Repair Response Message Format

5.3.3.7.3 ~~5.3.3.7.3~~ File Repair Response for Broadcast/Multicast of Repair Data

Details of how a file repair server decides, or is instructed, to use broadcast/multicast repair instead of point-to-point over HTTP are implementation specific and beyond the scope of the present document.

Prior to the decision to use broadcast/multicast repair, each repair response shall be provided by HTTP according to section 5.3.3.7.2.

The file repair server uses the HTTP response status code 302 (Found - Redirection) to indicate to the ~~terminalFD-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 an different (possibly new) download session may be defined for the repair.

The SDP file for broadcast/multicast repair session may be carried as payload (entity-body) in the HTTP response - which is especially useful if the broadcast/multicast repair session is a new (or recently end time modified) FLUTE download session and other means of service announcement prior to this were not feasible.

The delivery method's associatedDeliveryProcedureDescription may be updated and the new version transmitted in-band with the download session so that currently active client back-off timers are reset, thus minimizing additional ~~TerminalFD-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 can not be assured in all cases.

The existence of a broadcast/multicast file repair session is signalled by the inclusion of the optional *bmFileRepair* procedure in the updated Associated Delivery procedure description. This is signalled by the *bmFileRepair* element with a single "sessionDescriptionURI" attribute of type "xs:anyURI" which specifies the URI of the broadcast/multicast file repair session's session description.

In the cases where the same IP addressing is used for the broadcast/multicast repair session as the original download session, the ~~terminalFD-C~~ simply ~~shall not~~ **SHALL NOT** leave the group. Otherwise, the ~~terminalFD-C~~ ~~shall~~ **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 an BCAST download session, and the determination of end of files and session, and use of further associated delivery procedures uses the same techniques as specified for the BCAST download delivery method.

5.3.3.8 ~~5.3.3.8~~ Server Not Responding Error Case

In the error case where a ~~terminalFD-C~~ determines that the its selected file repair server is not responding it shall return to the serverURI list of repair servers and uniformly randomly select another server from the list, excluding any servers it has determined are not responding. All the repair requests message(s) from that ~~terminalFD-C~~ ~~shall~~ **SHALL** then be immediately sent to the newly selected file repair server.

If all of the repair servers from the serverURI list are determined to be not responding, the ~~terminalFD-C~~ ~~may~~ **MAY** attempt an HTTP GET to retrieve a, potentially new, instance of the session's Associated Procedure Description; otherwise ~~terminalFD-C~~ behaviour in this case is unspecified.

A ~~terminalFD-C~~ determines that a file repair server is not responding if any of these conditions apply:

1. The ~~terminalFD-C~~ is unable to establish a TCP connection to the server.
2. The server does not respond to any of the HTTP repair requests that have been sent by the ~~terminalFD-C~~ (it is possible that second and subsequent repair requests are sent before the first repair request is determined to be not-responded-to).

3. The server returns an unrecognized message (not a recognizable HTTP response).
The server returns an HTTP server error status code (in the range 500 to 505).

5.4 File Distribution over back-end interfaces

~~<TBD>~~

5.3.4 XML Schema for Associated Delivery Procedures

5.3.4.1 Generic Associated Delivery Procedure Description

Below is the formal XML syntax of associated delivery procedure description instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="associatedProcedureDescription" type="associatedProcedureType"/>
  <xs:complexType name="associatedProcedureType">
    <xs:sequence>
      <xs:element name="postFileRepair" type="basicProcedureType" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="bmFileRepair" type="bmFileRepairType" minOccurs="0" maxOccurs="1"/>
      <xs:element name="postReceptionReport" type="reportProcedureType" minOccurs="0"
        maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="basicProcedureType">
    <xs:sequence>
      <xs:element name="serverURI" type="xs:anyURI" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="offsetTime" type="xs:unsignedLong" use="optional"/>
    <xs:attribute name="maxBackOff" type="xs:unsignedLong" use="required"/>
  </xs:complexType>

  <xs:complexType name="bmFileRepairType">
    <xs:attribute name="sessionDescriptionURI" type="xs:anyURI" use="required"/>
  </xs:complexType>
  <xs:complexType name="reportProcedureType">
    <xs:simpleContent>
      <xs:extension base="basicProcedureType">
        <xs:attribute name="samplePercentage" type="xs:decimal" default="100" use="optional"/>
        <xs:attribute name="forceTimeIndependence" type="xs:boolean" default="false"
          use="optional"/>
        <xs:attribute name="reportType" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>

"reportType" value = "rack" || "star" || "star-all"
```

5.3.4.2 XML Syntax for a Reception Report Request

Below is the formal XML syntax of reception report request instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="receptionReport">
    <xs:choice>
      <xs:element name="receptionAcknowledgement" type="rackType"/>
    </xs:choice>
  </xs:element>
</xs:schema>
```



```

</xs:element name="statisticalReport" type="starType"/>
</xs:choice>
</xs:element>
<xs:complexType name="rackType">
  <xs:sequence>
    <xs:element name="fileURI" type="xs:anyURI"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="starType">
  <xs:simpleContent>
    <xs:element name="fileURI" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded">
      <xs:attribute name="receptionSuccess" type="xs:boolean" use="optional"/>
    </xs:element>
    <xs:attribute name="sessionId" type="xs:string" use="optional"/>
    <xs:attribute name="sessionType" type="xs:string" use="optional"/>
    <xs:attribute name="serviceId" type="xs:string" use="optional"/>
    <xs:attribute name="clientId" type="xs:string" use="optional"/>
    <xs:attribute name="serverURI" type="xs:anyURI" use="optional"/>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>

```

5.3.4.3 Use of Specific Values

"sessionType" value = {"download", "streaming", "mixed"}

6. Stream Distribution

~~6.1 Functional Architecture of Stream Distribution~~

~~6.1.1 General~~

text

text

~~6.1.1.1 Example Level 4~~

text

~~6.2 Protocol for Stream Distribution~~

6.2.16.1

roduction

Int

<text>

6.2.26.2

P as Transport Protocol

RT

text The Real-Time Transport Protocol [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 RTCP [note : RFC to be added] packets in the downlink (sender reports) is allowed, but not required. 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 MAY send RTCP packets (sender reports). If synchronization is required the sender SHALL send RTCP packets (sender reports)
- The terminal SHALL support 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)

6.3 ~~6.2.3~~ RTP payload formats

RTP can in general transport any audiovisual data, if an RTP payload format is defined for it. BCAST Stream Distribution function allows the use of any RTP payload format. The payload formats for each Broadcast Distribution System are defined in the normative specifications associated with each Broadcast Distribution System as outlined in the respective BDS Adaptation specifications of BCAST.~~For RTP/UDP/IP transport of continuous media the following RTP payload formats SHALL be supported:~~

~~—TBD~~

6.2.2.1 Example Level 4

~~<text>~~

6.3 Procedures for Stream Distribution

6.3.1 Procedure 1

~~<Figure> and <text>~~

6.3.2 Procedure 2

~~<Figure> and <text>~~

6.3.3 Procedure

~~<Figure> and <text>~~

~~7. Media Codecs and Formats [Informative]~~

~~7.1 General~~

~~7.2 Speech~~

~~7.3 Audio~~

~~7.4 Video~~

~~7.5 Still Images~~

~~7.6 Text~~

~~7.7 Other Media Codecs and Fortmats~~

8. ~~Supplementary methods and Procedures for File Distribution and Stream Distribution~~

8.1 ~~General~~

8.2 ~~Error resilience~~

8.2.1 ~~Methods for Error resilience~~

8.2.1.1 ~~Pre-Error resilience Methods~~

8.2.1.2 ~~Point to Point Error Recovery~~

8.3 ~~Error reporting~~

8.4 ~~Others~~

Appendix A. Change History

(Informative)

<< The following is a model of a revision table. DELETE THIS COMMENT >>

A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA
OMA-xyyz-V1_0-20021001-A	01 Oct 2002	Initial document to address the basic starting point Ref TP Doc# OMA-TP-2002-1234-xyyzForApproval
OMA-xyyz-V1_1-20030405-A	05 Apr 2003	Description of changed Ref TP Doc# OMA-TP-2003-0321-xyyzV1_1forApproval

A.2 Draft/Candidate Version <current version> History

<< This section is available in pre-approved versions - it should be removed in the actual approved versions. DELETE THIS COMMENT >>

Document Identifier	Date	Sections	Description
Draft Versions OMA-TS-BCAST-DistributionV1_0_0-20050106-D	06 Jan 2005	5,6,7, and 8	Agreed Draft
OMA-TS-BCAST-DistributionV1_0_0-20050607-D	07 Jun 2005	2.1 and 5.2.1	CR agreed during BCAST Conference Call on June 1 st : OMA-BCAST-2005-0036R6-CR-File-Delivery
OMA-TS-BCAST-DistributionV1_0_0-20050912-D	12 SEP 2005	5.3, 6.2.2 and 6.2.3	CRs agreed during BCAST Montreal Meeting st : OMA-BCAST-2005-0362R1-CR-Stream-Delivery-Protocol OMA-BCAST-2005-0363R2-CR-Associated-Delivery-procedure
OMA-TS-BCAST-DistributionV1_0_0-20051107-D	07 Nov 2005	6.2.2	CRs agreed during BCAST Tokyo Meeting OMA-BCAST-2005-0448R1-CR-Distr-RTP-Normative
Candidate Version OMA-xyyz-V1_2	16 Sep 2003	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2003-0abc-CandidateRequest_xyyz_V1_2
Draft Version OMA-xyyz-V1_2	24 Sep 2003	6.8	Status changed to Draft (demoted) to address important class 1 CR OMA-XY-2003-0172-CR_AddSectionOnJellyGoesOnTop
Candidate Versions OMA-xyyz-V1_2	13 Nov 2003	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2003-0def-CandidateRequest_xyyz_V1_2_again
	21 Dec 2003	4.2, 6.3	Minor CR to address interpretation of bread references OMA-XY-2003-0205-CR_SlicedBreadClarification Notice sent to TP of minor update TP ref # OMA-TP-2003-0ghi-CandidateUpdateNotice_xyyz_V1_2
	12 Jan 2004	4.2, 6.6	Minor CR to cover cases where knife not available OMA-XY-2004-0012-CR_SpreadingWithoutKnife Notice sent to TP of minor update TP ref # OMA-TP-2004-0jkl-CandidateUpdateNotice_xyyz_V1_2

Appendix B. Static Conformance Requirements (Normative)

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

The following is a model of a set of SCR tables. DELETE THIS COMMENT

B.1 SCR for XYZ Client

Item	Function	Reference	Status	Requirement
XYZ-C-001	Something mandatory	Section x.y	M	(XYZ-C-001 OR XYZ-C-003) AND XYZ-C-002
XYZ-C-002	Something optional	Section x.y	O	
XYZ-C-003	Dependencies on ZYX	Section x.y	M	ZYX:MCF
XYZ-C-004	Dependencies on ZYX	Section x.y	O	ZYX:OCF

B.2 SCR for XYZ Server

Item	Function	Reference	Status	Requirement
XYZ-S-001	Something mandatory	Section x.y	M	XYZ-S-001 OR XYZ-S-002 OR XYZ-S-003
XYZ-S-002	Something optional	Section x.y	O	
XYZ-S-003	Dependencies on ZYX	Section x.y	M	ZYX:MSF
XYZ-S-004	Dependencies on ZYX	Section x.y	O	ZYX:OSF

Appendix C. XML-Schema for Associated Delivery Procedures <Additional Information> (Normative)

~~If needed, add annex to provide additional information to support the document. In general, this information should be informative, as normative material should be contained in the primary body of the document.~~

~~Note that the styles for the headers in the appendix (App1, App2, App3) are different than the main body. The use below is intended to validate the styles to be used. Remove if not needed.~~

~~DELETE THIS COMMENT~~

C.1 Generic Associated Delivery Procedure Description App Headers

Below is the formal XML syntax of associated delivery procedure description instances.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="associatedProcedureDescription" type="associatedProcedureType"/>
  <xs:complexType name="associatedProcedureType">
    <xs:sequence>
      <xs:element name="postFileRepair" type="basicProcedureType" minOccurs="0"
        maxOccurs="1"/>
      <xs:element name="bmFileRepair" type="bmFileRepairType" minOccurs="0" maxOccurs="1"/>
      <xs:element name="postReceptionReport" type="reportProcedureType" minOccurs="0"
        maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="basicProcedureType">
    <xs:sequence>
      <xs:element name="serverURI" type="xs:anyURI" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="offsetTime" type="xs:unsignedLong" use="optional"/>
    <xs:attribute name="maxBackOff" type="xs:unsignedLong" use="required"/>
  </xs:complexType>
  <xs:complexType name="bmFileRepairType">
    <xs:attribute name="sessionDescriptionURI" type="xs:anyURI" use="required"/>
  </xs:complexType>
  <xs:complexType name="reportProcedureType">
    <xs:simpleContent>
      <xs:extension base="basicProcedureType">
        <xs:attribute name="samplePercentage" type="xs:decimal" default="100" use="optional"/>
        <xs:attribute name="forceTimeIndependence" type="xs:boolean" default="false"
          use="optional"/>
        <xs:attribute name="reportType" type="xs:string" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>

"reportType" value = "rack" || "star" || "star-all"
```

C.2 XML Syntax for a Reception Report Request

Below is the formal XML syntax of reception report request instances.


```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="receptionReport">
    <xs:choice>
      <xs:element name="receptionAcknowledgement" type="rackType"/>
      <xs:element name="statisticalReport" type="starType"/>
    </xs:choice>
  </xs:element>
<xs:complexType name="rackType">
  <xs:sequence>
    <xs:element name="fileURI" type="xs:anyURI"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="starType">
  <xs:simpleContent>
    <xs:element name="fileURI" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded">
      <xs:attribute name="receptionSuccess" type="xs:boolean" use="optional"/>
    </xs:element>
    <xs:attribute name="sessionId" type="xs:string" use="optional"/>
    <xs:attribute name="sessionType" type="xs:string" use="optional"/>
    <xs:attribute name="serviceId" type="xs:string" use="optional"/>
    <xs:attribute name="clientId" type="xs:string" use="optional"/>
    <xs:attribute name="serverURI" type="xs:anyURI" use="optional"/>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>

```

C.3 Use of Specific Values

"sessionType" value = {"download", "streaming", "mixed"}

<More text>

C.1.1 More Headers