Video Specifications
IP Multicast

IP Multicast Controller-Server Interface Specification

OC-SP-MC-MSI-C01-161026

CLOSED

DISCLAIMER

This document is furnished on an "AS IS" basis and neither CableLabs nor its members provides any representation or warranty, express or implied, regarding the accuracy, completeness, noninfringement, or fitness for a particular purpose of this document, or any document referenced herein. Any use or reliance on the information or opinion in this document is at the risk of the user, and CableLabs and its members shall not be liable for any damage or injury incurred by any person arising out of the completeness, accuracy, or utility of any information or opinion contained in the document.

CableLabs reserves the right to revise this document for any reason including, but not limited to, changes in laws, regulations, or standards promulgated by various entities, technology advances, or changes in equipment design, manufacturing techniques, or operating procedures described, or referred to, herein.

This document is not to be construed to suggest that any company modify or change any of its products or procedures, nor does this document represent a commitment by CableLabs or any of its members to purchase any product whether or not it meets the characteristics described in the document. Unless granted in a separate written agreement from CableLabs, nothing contained herein shall be construed to confer any license or right to any intellectual property. This document is not to be construed as an endorsement of any product or company or as the adoption or promulgation of any guidelines, standards, or recommendations.

Document Status Sheet

Document Control Number	OC-SP-MC-MSI-C01-161026			
Document Title	IP Multicast Controller-Server Interface Specification			
Revision History	I01 - 05/28/2015 C01 - 10/26/2016			
Date	October 26, 2016			
Status	~~Work in Progress~~	~~Draft~~	~~Issued~~	**Closed**
Distribution Restrictions	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Key to Document Status Codes

Work in Progress An incomplete document, designed to guide discussion and generate feedback that may include several alternative requirements for consideration.

Draft A document in specification format considered largely complete, but lacking review by Members and vendors. Drafts are susceptible to substantial change during the review process.

Issued A generally public document that has undergone Member and Technology Supplier review, cross-vendor interoperability, and is for Certification testing if applicable. Issued Specifications are subject to the Engineering Change Process.

Closed A static document, reviewed, tested, validated, and closed to further engineering change requests to the specification through CableLabs.

Trademarks

Contents

Figures

Tables

This page left blank intentionally.

1 SCOPE

1.1 Overview

This specification is part of the Video family of specifications developed by Cable Television Laboratories, Inc. (CableLabs) and published under the OpenCable License Agreement. The IP Multicast MC-MS Interface Specification defines an interface identified in [IPM-TR] between the Multicast Controller and the Multicast Server. The intent of this specification is to provide multi-vendor interoperability across this interface such that interoperable products can be brought to market which support Multicast-ABR (Adaptive Bit Rate).

The IP Multicast specifications primarily adopt web services as the standard communications mechanism between components.

1.2 Purpose

This document specifies the interface between the Multicast Controller and the Multicast Server.

1.3 Scope

This specification details the usage of HTTP and web services. The information in this specification applies to all of the web service interfaces defined for IP Multicast:

- Multicast Controller to Embedded Multicast Client (mc-emc) interface: The mc-emc interface is defined between a Multicast Controller in an operator's back office and a Multicast Client embedded in a residential gateway. This interface is used to signal viewing-related activity and the list of content streams available via IP multicast.
- Multicast Controller to Multicast Server (mc-ms) interface: The mc-ms interface is defined between the Multicast Controller, which controls what streams are available on multicast and when, and the Multicast Server, which performs content retrieval and multicast delivery.

The interfaces defined in the IP Multicast reference architecture are shown in Figure 1.

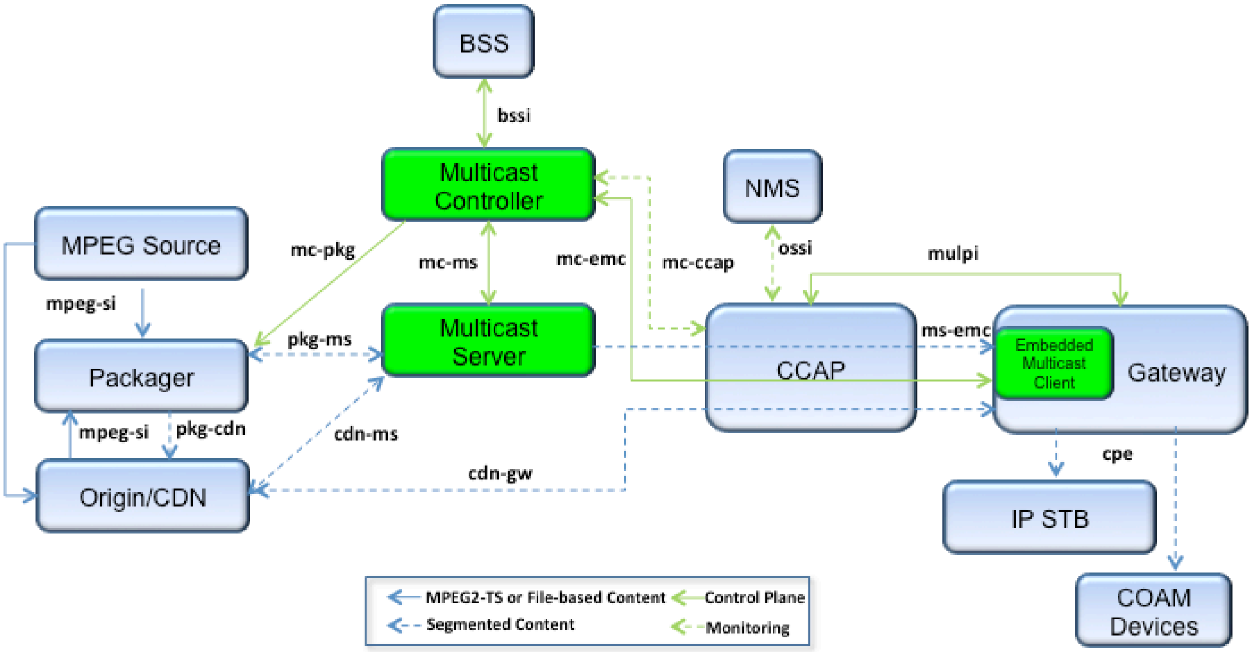

Figure 1 - IP Multicast Reference Architecture

1.4 Requirements

Throughout this document, the words that are used to define the significance of particular requirements are capitalized. These words are:

"MUST" This word means that the item is an absolute requirement of this specification.

"MUST NOT" This phrase means that the item is an absolute prohibition of this specification.

"SHOULD" This word means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before choosing a different course.

"SHOULD NOT" This phrase means that there may exist valid reasons in particular circumstances when the listed behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

"MAY" This word means that this item is truly optional. For example, one vendor may choose to include the item because a particular marketplace requires it or because it enhances the product; another vendor may omit the same item.

This document defines many features and parameters, and a valid range for each parameter is usually specified. Equipment requirements are always explicitly stated. Equipment is required to comply with all mandatory (MUST and MUST NOT) requirements to be considered compliant with this specification. Support of non-mandatory features and parameter values is optional.

2 REFERENCES

2.1 Normative References

This specification uses the following normative reference.

[MC-EMC] IP Multicast Controller-Client Interface Specification, OC-SP-MC-EMCI-C01-161026, October 26, 2016, Cable Television Laboratories, Inc.

[MS-EMC] IP Multicast Server-Client Interface Specification, OC-SP-MS-EMCI-C01-161026, October 26, 2016, Cable Television Laboratories, Inc.

2.2 Informative References

This specification uses the following informative references.

[IPM-TR] IP Multicast Assisted Bit Rate Architecture Technical Report, OC-TR-IP-MULTI-ARCH-C01-161026, October 26, 2016, Cable Television Laboratories, Inc.

[RFC 2616] IETF RFC 2616, Hypertext Transfer Protocol - HTTP/1.1, June 1999.

[RFC 3376] IETF RFC 3376, Internet Group Management Protocol, Version 3, October 2002.

[RFC 3453] IETF RFC 3453, The Use of Forward Error Correction (FEC) in Reliable Multicast, December 2002.

[RFC 5052] IETF RFC 5052, Forward Error Correction (FEC) Building Block, August 2007.

[RFC 5401] IETF RFC 5401, Multicast Negative-Acknowledgment (NACK) Building Blocks, November 2008.

[RFC 5510] IETF RFC 5510, Reed-Solomon Forward Error Correction (FEC) Schemes, April 2009.

[RFC 5740] IETF RFC 5740, NACK-Oriented Reliable Multicast (NORM) Transport Protocol, November 2009.

[RMI HTTP] Resource Management Architecture and HTTP Specification, CM-SP-RMI-HTTP-I02-150528, May 28, 2015, Cable Television Laboratories, Inc.

2.3 Reference Acquisition

- Cable Television Laboratories, Inc., 858 Coal Creek Circle, Louisville, CO 80027;
  Phone +1-303-661-9100, Fax +1-303-661-9199; http://www.cablelabs.com/
- Internet Engineering Task Force (IETF): http://www.ietf.org/

3 TERMS AND DEFINITIONS

This specification uses the following terms:

Access Network The HFC network between the Gateway and the CCAP.

Adaptive Bit Rate A streaming video technique where Players select between multiple bit rate encodings of the same video stream.

Bonding Group A logical set of DOCSIS channels which support parallel transmission

Companion Device A video playback device which is not a television such as a tablet, smartphone or PC.

Converged Cable Access Platform A system which provides DOCSIS and QAM-based video services to CMs, Gateways and set-top boxes.

Content Distribution Network A network designed to minimizing latency by distributing network objects onto geographically diverse servers.

Embedded Multicast Client The function embedded in the Gateway which joins multicast groups and receives multicast content.

Gateway A customer premises device which facilitates delivery of video, data and other services.

Headend The central location on the cable network that is responsible for injecting broadcast video and other signals in the downstream direction.

Home Network A network within the subscriber premises which connects to the Access Network via the Gateway.

IP Multicast A delivery mechanism whereby IP packets can be transmitted to/received from devices that have explicitly joined a multicast group.

Key Server A server which provides keys as part of a DRM solution.

License Server A server which checks authorization and provides licenses as part of a DRM solution.

Linear TV A continuous content stream from a provider, e.g., a broadcast television network.

MPEG Source A device which provides a source of MPEG-encoded video content for encoding as ABR content streams.

Multicast Controller A device which controls what channels are provided via multicast.

Multicast Server A device which delivers content via multicast.

Multiple-System Operator (MSO) A company that owns and operates more than one cable system.

Packager A device which takes continuous video streams, encodes them at different bit rates and breaks them into shorter duration segments.

PacketCable Multimedia An application agnostic QoS architecture for services delivered over DOCSIS networks.

Player An application for playback of ABR video.

Serving Group A set of receivers which all receive the same transmission of a given frequency band.

Stream A series of video segments which contain the same video asset, typically at the same bit rate encoding.

Unicast Delivery of IP packets to a single device.

4 ABBREVIATIONS AND ACRONYMS

This specification uses the following terms:

ABR	Adaptive Bit Rate
BSS	Business Support System
CCAP	Converged Cable Access Platform
CDN	Content Delivery Network
CM	Cable Modem
CMS	Content Management Server
COAM	Customer Owned and Managed
CPE	Customer Premises Equipment
DNS	Domain Name System
DOCSIS®	Data-Over-Cable Service Interface Specifications
EAS	Emergency Alert System
EAN	Emergency Action Notification
FEC	Forward Error Correction
GW	Gateway
HD	High Definition
HDS	HTTP Dynamic Streaming
HLS	HTTP Live Streaming
HTTP	Hyper Text Transfer Protocol
IGMP	Internet Group Management Protocol
IP	Internet Protocol
IPsec	Internet Protocol Security
IP-STB	IP Set-top Box
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
JSON	JavaScript Object Notation
M-ABR	Multicast-Adaptive Bit Rate
MC	Multicast Controller
MLD	Multicast Listener Discovery
MoCA	Multimedia over Coax Alliance
MPEG	Moving Picture Experts Group
MPEG-DASH	Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP
MS	Multicast Server
MSS	Microsoft Smooth Streaming
NACK	Negative-Acknowledgement
NMS	Network Management System
NORM	NACK-Oriented Reliable Multicast
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service

REST	Representational State Transfer
RTP	Real-time Transport Protocol
RTCP	RTP Control Protocol
RTSP	Real-Time Streaming Protocol
RTMP	Real-Time Messaging Protocol
SD	Standard Definition
SDV	Switched Digital Video
(S,G)	(Source IP Address, Group IP Address)
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TR	Technical Report
UA	User Agent
UDP	User Datagram Protocol
UE	User Equipment
URI	Uniform Resource Identifier
WiFi	Wireless Local Area Network
XML	eXtensible Markup Language

5 OVERVIEW AND THEORY OF OPERATIONS

Multicast-ABR or, perhaps more accurately, Multicast-assisted ABR is a technique for using IP multicast to deliver ABR video segments to a Gateway. It is a network-layer efficiency mechanism which is transparent to ABR Players.

Multicast-ABR is more fully described in [IPM-TR] and Section 5.2.

5.1 Design Principles

This protocol was designed to provide centralized control of the streams available via multicast as well as the multicast streams watched (or, more precisely, joined) by individual Gateways. This was done to minimize complexity on the Gateway for reduced cost and increased service agility.

This protocol is uses a web services architecture. The web services API structure is based on [RMI HTTP] as this protocol is used to deliver other video services.

5.2 Functional Overview

5.2.1 M-ABR Infrastructure Interactions

The following sequence diagram is intended to illustrate a general data flow utilized on the MC-MS interface.

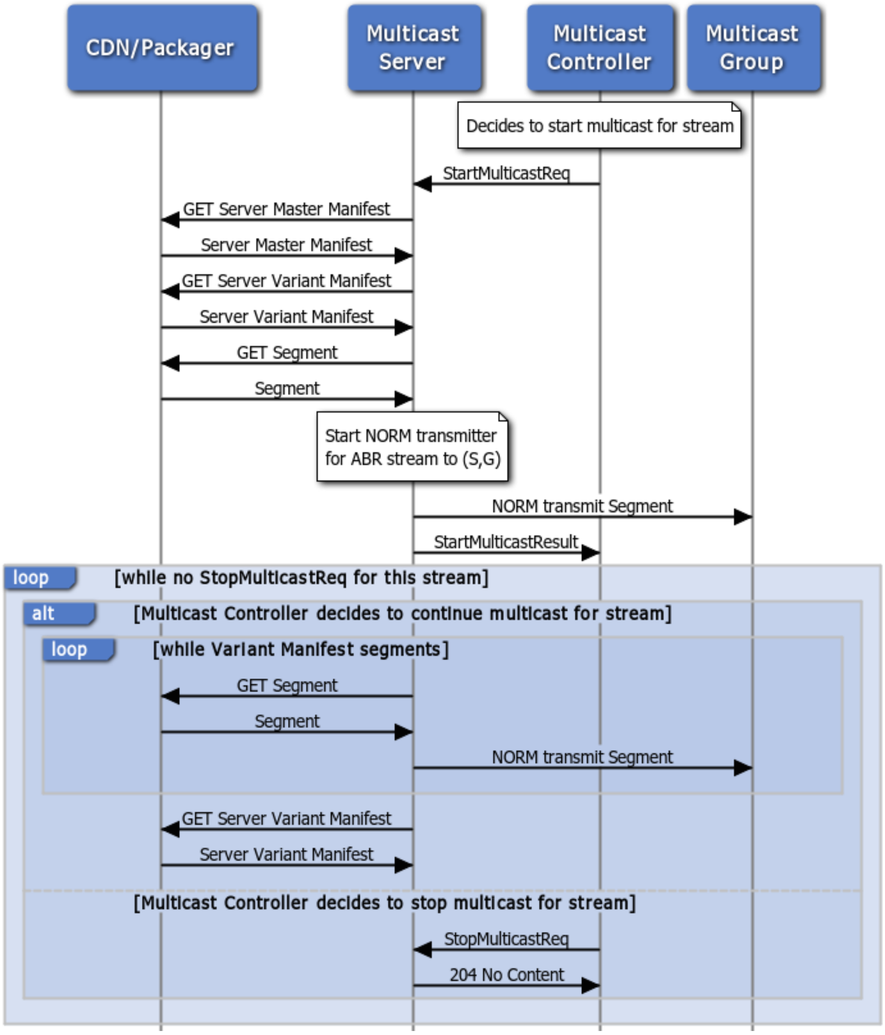
CDN/Packager
Multicast Server
Multicast Controller
Multicast Group
Decides to start multicast for stream
StartMulticastReq
GET Server Master Manifest
Server Master Manifest
GET Server Variant Manifest
Server Variant Manifest
GET Segment
Segment
Start NORM transmitter for ABR stream to (S,G)
NORM transmit Segment
StartMulticastResult
loop
[while no StopMulticastReq for this stream]
alt
[Multicast Controller decides to continue multicast for stream]
loop
[while Variant Manifest segments]
GET Segment
Segment
NORM transmit Segment
GET Server Variant Manifest
Server Variant Manifest
[Multicast Controller decides to stop multicast for stream]
StopMulticastReq
204 No Content

Figure 2 - Example High-Level Infrastructure Interactions

The Multicast Controller dictates what content and bit rates are delivered via multicast. This is communicated to the Multicast Server via the StartMulticastReq message. Once the Multicast Server knows the content to multicast, it can get that content. Some architectures have the MS get the content from the Packager and others from the CDN, but the result is the same - the Multicast Server acquires video segments for the stream, typically via HTTP almost as though it were a Player.

Different operators use different mechanisms to determine what content to multicast. In some architectures, the BSS drives the set of content which is multicast (this is referred to as Policy-Driven Multicast in [IPM-TR]). In other architectures the content to multicast is determined more dynamically based on actual viewership requests (this is referred to as Viewership-Driven Multicast in [IPM-TR]). However, the protocol mechanisms used by the Multicast Controller to direct the Multicast Server which content streams to multicast when (as defined by this specification) are independent of these policies and Multicast Controller implementation decisions.

The Multicast Server continues to acquire video segment files and transmit them via NORM (NACK-Oriented Reliable Multicast) over the MS-EMC interface for a given video stream until the Multicast Controller directs it to stop via a StopMulticastReq message.

6 MC-MS PROTOCOL OPERATION

The Multicast Controller contains policy configuration and real-time information from Gateways which it uses to determine which ABR streams to provide on multicast at any given time. The Multicast Controller can also utilize capacity-related information such as the number of active multicast streams or the aggregate multicast bit rate to inform its decisions on how many content streams should be provided via multicast. The specific details on how these decisions are made are outside the scope of this specification.

6.1 Protocol Operation

6.1.1 Start Multicast Stream

To start a multicast stream of ABR video segments, the Multicast Controller MUST POST a StartMulticastReq message to the /ms/multicast URL of the Multicast Server. Upon receipt of a StartMulticast message, the Multicast Server MUST perform the following steps:

1. If a sourceAddress is included, verify that the Multicast Server can bind to that address
2. Retrieve the master/top-level manifest for the stream
3. Verify that the master/top-level contains the bit rate commanded by the Multicast Controller
4. Retrieve the variant manifest for the requested bit rate
5. Retrieve the first segment found in the variant manifest
6. Transmit the first segment on the directed multicast group and port (and, if present, sourceAddress)

If all of these steps are successful, the Multicast Server MUST return 200 OK as its HTTP response code and include a StartMulticastResult. Otherwise, the Multicast Server returns an HTTP response code with an appropriate error value and a Response message body with more detailed information on the failure.

If the StartMulticast message does not include a sourceAddress, the Multicast Server MUST assign a sourceAddress to the (S,G) utilized for multicasting this stream. If the StartMulticast message does not include a sourceAddress, the Multicast Server MUST return the assigned sourceAddress in the StartMulticastResult message returned to the Multicast Controller.

6.1.2 Stop Multicast Stream

If the Multicast Controller decides to stop an active multicast session, the Multicast Controller MUST send the Multicast Server a StopMulticast request. Upon receipt of a StopMulticast request, the Multicast Server MUST look for a matching stream in its current multicast stream set. If a match is found, the Multicast Server MUST stop retrieving segments for this stream and MUST stop sending these segments via multicast. Upon successful completion of these steps, the Multicast Server MUST return 204 No Content as its HTTP response code. Otherwise, the Multicast Server returns an HTTP response code with an appropriate error value and a Response message body with more detailed information on the failure.

6.1.3 Status

It is possible that the Multicast Controller will want to compare its state against the state of a given Multicast Server. In this case, the Multicast Controller MAY send a GET request to the /ms/multicast URL with or without a sessionId parameter. (Other management systems might want to use this method as well.)

If the Multicast Server receives GET request on the /ms/multicast URL with a sessionId for a known session, it MUST respond with a MulticastStatusResult message containing the status of the referenced multicast session. If the Multicast Server receives GET request on the /ms/multicast URL with no sessionId, it MUST respond with a MulticastStatusListResult message containing the complete set of streams it is currently attempting to retrieve and transmit via multicast. In the event of an error, the Multicast Server MUST send an appropriate HTTP response code for the error as well as a Response message body with more detailed information on the failure.

The Multicast Server SHOULD include the bytesSent and lastSegmentFileSent attributes for each sessionId in its MulticastStatusResult and MulticastStatusListResult messages. If a given multicast session is errored, the Multicast Server MUST include the errorMessage and errorTime attributes in the Status element for that sessionId.

6.1.4 Send Multicast Channel Map

The Multicast Controller controls the content being multicast and the corresponding channel map; however, the Multicast Controller does not generally perform multicast transmissions. Thus, if an operator wants to use a multicast channel map, then often that operator needs to configure the Multicast Controller to send channel maps to the Multicast Server for multicast transmission. (Note: these channel maps are not intended to be processed by the Multicast Server; rather they are simply multicast by the Multicast Server as received from the Multicast Controller.)

The Multicast Controller MAY POST a SendChannelMapReq message to the Multicast Server. If the Multicast Server receives an SendChannelMapReq message, it transmits the ChannelMap element of the message to the multicast group specified in the message using NORM per [MS-EMC]. The Multicast Server SHOULD wait until it has transmitted the resulting ChannelMapMsg via NORM before sending a SendChannelMapResult message back to the Multicast Controller.

If the SendChannelMapReq message does not include a sourceAddress, the Multicast Server MUST assign a sourceAddress to the (S,G) utilized for multicasting this stream. If the SendChannelMapReq message does not include a sourceAddress, the Multicast Server MUST return the assigned sourceAddress in the SendChannelMapResult message returned to the Multicast Controller.

7 MC-MS INTERFACE DEFINITION

This section defines the interface and protocol between the Multicast Controller (MC) and the Multicast Server (MS). In general, the MC specifies the content to be delivered via multicast while the MS performs the actual multicast delivery of content.

7.1 Multicast Stream Functions

7.1.1 StartMulticast

To initiate transmission of a series of ABR segments to a given multicast group, the MC transmits a StartMulticast message to the MS.

Request Direction: MC to MS

Method: HTTP POST

Message URL: `http://<device-url>/ms/multicast`

Children: None

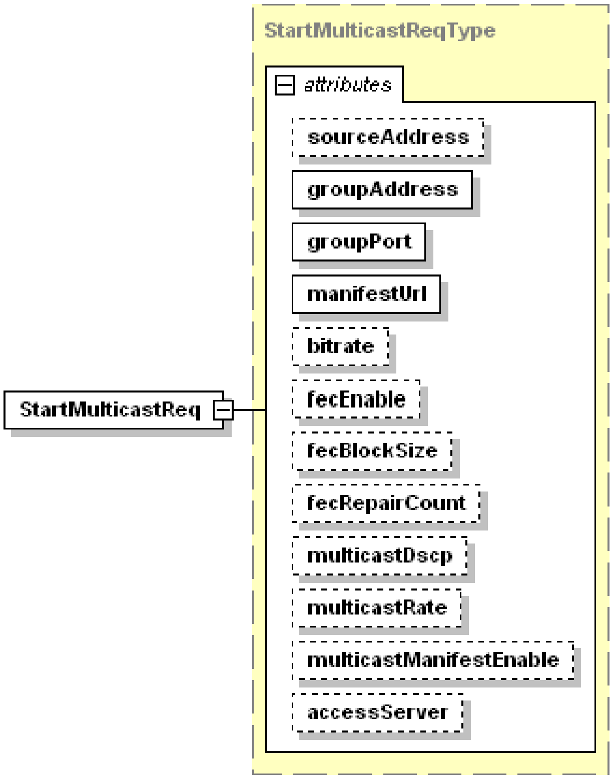

Figure 3 - StartMulticastStreamReqMsg

Table 1 - StartMulticast Attribute Definitions

Attribute	Use	Data Type	Description
sourceAddress	Optional	dvb:IPOrDomainType	The source address the MC instructs the MS to use for this (S,G). If omitted, the MS selects the sourceAddress.
groupAddress	Required	dvb:IPOrDomainType	The address the MC instructs the MS to use as the destination address for the multicast stream.

Attribute	Use	Data Type	Description
groupPort	Required	xs:short	The port the MC instructs the MS to use as the destination port for the multicast stream.
manifestUrl	Required	xs:string	The URL of a manifest of the stream to multicast. If the 'bitrate' attribute is present, this is the URL of a master manifest, and the MS uses the 'bitrate' attribute to identify the appropriate variant manifest URL within the master manifest. If the 'bitrate' attribute is not present, this is the URL of the variant manifest the MS uses to retrieve stream segments.
bit rate	Optional	xs:string	Bit rate of the stream entry in the manifest that the MS is instructed to select. For this transaction to be successful, this needs to match a BANDWIDTH attribute in an EXT-X-STREAM-INF tag. Note: this is encoded as a stream to support older encodings such as "WiFi". This attribute can also be used to encode a bit rate index.
fecEnable	Optional	xs:boolean	Enable/disable NORM FEC for this stream.
fecBlockSize	Optional	xs:unsignedByte	NORM FEC block size (k) for this stream.
fecRepairCount	Optional	xs:unsignedByte	NORM FEC repair count (n_r) for this stream.
multicastDscp	Optional	xs:unsignedByte	The DSCP byte to use on NORM traffic for this stream.
multicastRate	Optional	xs:unsignedInt	The transmission rate for this stream. The Multicast Server MUST treat encodings where the 'multicastRate' is less than the 'bitrate' for a given stream as an error.
accessServer	Optional	dvb:IPOrDomainType	The address the MC instructs the MS to use for retrieving CDN tokens.

7.1.2 StartMulticast Response

On successful completion of a StartMulticast request, the Multicast Server MUST return 200 OK with an XML response carrying the StartMulticastResult element, detailing the selected sessionId and other optional information. The StartMulticastResult element is defined in Section 7.1.2.1.

On the failure of a StartMulticast request on the MS, the Multicast Server MUST return 500 Internal Server Error and provide a StartMulticastResult element with details about the failure in its Response element.

On a StartMulticast request that leads to a standard HTTP protocol error, the Multicast Server MUST return the appropriate HTTP status code and status text.

Table 2 - StartMulticast Response Status Codes

Status Code	Meaning
200	OK
400	Bad Request
500	Internal Server Error

7.1.2.1 StartMulticastResult

The StartMulticastResult element is returned with the XML response to a StartMulticast request from the Multicast Controller when no HTTP errors occur. It provides the status of the response, details of any errors that occurred and the resulting impact of the request on the Multicast Server.

Children: Response(1)

StartMulticastDetails(0..1)

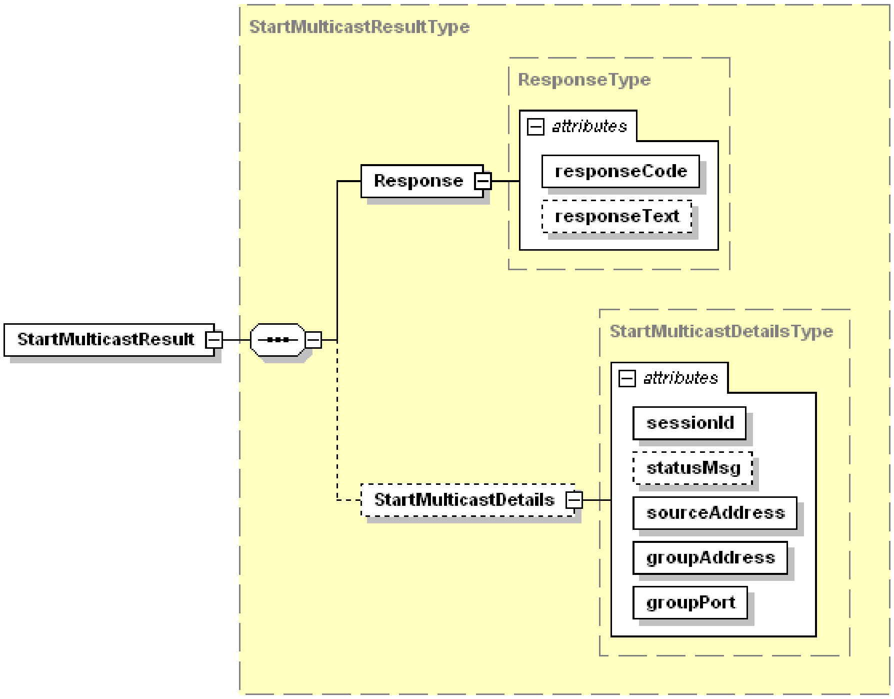

Figure 4 - StartMulticastResult

7.1.2.1.1 Response

The Response element is defined in [MC-EMC]. The Multicast Server MUST provide a response code value of 200 on a successful completion of the StartMulticast request. Standard StartMulticast failure response codes are detailed in the Response Codes section of [MC-EMC].

7.1.2.1.2 StartMulticastDetails

Table 3 - StartMulticastDetails Attribute Definitions

Attribute	Use	Data Type	Description
sessionId	Required	xs:string	A globally unique identifier for the multicast session initiated per the Multicast Controller request.
statusMsg	Optional	xs:string	Any status message associated with this setup.
sourceAddress	Required	dvb:IPOrDomainType	The source address the MS was instructed or selected to use for this (S,G).
groupAddress	Required	dvb:IPOrDomainType	The group address the MS was instructed to use for this (S,G).
groupPort	Required	xs:short	The port the MS was instructed to use as the destination port for the multicast stream.

7.1.3 Start Multicast Examples

7.1.3.1 StartMulticast Request Example

```
POST /ms/multicast HTTP/1.1
Host: ms
User-Agent: mc
Content-Type: application/xml
```

```
Content-Length: ...
<StartMulticastReq
    groupAddress="224.1.1.1"
    groupPort="12345"
    bit rate="3389000"
    manifestUrl="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_6713276793419826
163_HLS/AnEHD_HD_NAT_14710_0_6713276793419826163_HLS.m3u8"
/>
```

7.1.3.2 StartMulticast Response Examples

7.1.3.2.1 StartMulticast Response Example: Success

```
HTTP/1.1 200 OK
Server: ms 1.1
Cache-Control: no-cache
<StartMulticastResult>
    <Response responseCode=”200”/>
    <StartMulticastDetails
        sessionId="a7cb6288-7b1d-11e4-b116-123b93f75cba"
        sourceAddress="10.10.10.123"/>
</StartMulticastResult>
```

7.1.3.2.2 StartMulticast Response Example: Server Error

```
HTTP/1.1 500 Internal Server Error
Server: ms 1.1
Cache-Control: no-cache
<StartMulticastResult>
    <Response responseCode=“502”
        responseText=”Invalid sourceAddress”/>
</StartMulticastResult>
```

7.1.4 StopMulticast Request

To stop a multicast transmission session, the MC sends a POST to the /StopMulticast URL of the MS including the sessionId of the session to be stopped

Request Direction: MC to MS

Method: HTTP POST

Message URL: `http://<device-url>/ms/StopMulticast/<sessionId>`

Children: None

Table 4 - StopMulticast Parameter Definitions

Parameter	Use	Data Type	Description
sessionId	Required	xs:string	A globally unique identifier for the session to delete.

7.1.5 StopMulticast Response

On success the Multicast Server MUST return response status of 204 No Content.

If the Multicast Controller specifies an unknown sessionId, the Multicast Server MUST return a status code of 404 Not Found.

On standard HTTP protocol errors, the Multicast Server MUST return the appropriate HTTP status code and status text.

If the request fails due to reasons other than unknown sessionId or standard HTTP protocol errors, the Multicast Server MUST return a status code of 500 Internal Server Error and provide the StopMulticastResult element in the XML body with details about the failure in a Response element.

Table 5 - StopMulticast Response Status Codes

Status Code	Meaning
204	No Content
400	Bad Request
404	Not Found
500	Internal Server Error

7.1.5.1 StopMulticastResult

The StopMulticastResult element is returned with the XML response to a StopMulticast request from the Multicast Controller when an internal server error occurs. It provides, in the Response element, the details of the error that occurred. The StopMulticastResult element has no attributes.

Children: Response (1)

7.1.5.1.1 Response

The Response element is detailed in [MC-EMC]. Valid response codes are detailed in the Response Codes section of [MC-EMC].

7.1.6 Stop Multicast Examples

7.1.6.1 StopMulticast Request Example

```
POST /ms/StopMulticast/a7cb6288-7b1d-11e4-b116-123b93f75cba HTTP/1.1
Host: ms
User-Agent: mc
```

7.1.6.2 StopMulticast Response Example: Success

```
HTTP/1.1 204 No Content
Server: ms 1.1
Cache-Control: no-cache
```

7.1.6.3 StopMulticast Response Example: Invalid Session Id

```
HTTP/1.1 404 Not Found
Server: ms 1.1
Cache-Control: no-cache
```

7.1.6.4 StopMulticast Response Example: Internal Server Error

```
HTTP/1.1 500 Internal Server Error
Server: ms 1.1
Cache-Control: no-cache
<StopMulticastResult>
    <Response responseCode=“601”
        responseText=”Implementation Specific Error”/>
</StopMulticastResult>
```

7.2 Multicast Status Functions

7.2.1 Multicast Status Request

To determine the current state of a multicast transmission stream or streams, the MC sends a GET request to the /multicast URL of the MS.

Request Direction: MC to MS

Method: HTTP GET

Message URL: `http://<device-url>/ms/multicast{/sessionId}`

Children: None

Table 6 - MulticastStatus Parameter Definitions

Parameter	Use	Data Type	Description
sessionId	Optional	xs:string	A globally unique identifier for the session to check status on. If omitted, the Multicast Server reports on all sessions.

7.2.2 Multicast Status Response

The response to a MulticastStatus request can take one of two forms - a result for an individual stream or a result for all streams. The form returned is determined by whether or not the request URL includes the sessionId.

7.2.2.1 MulticastStatusResult

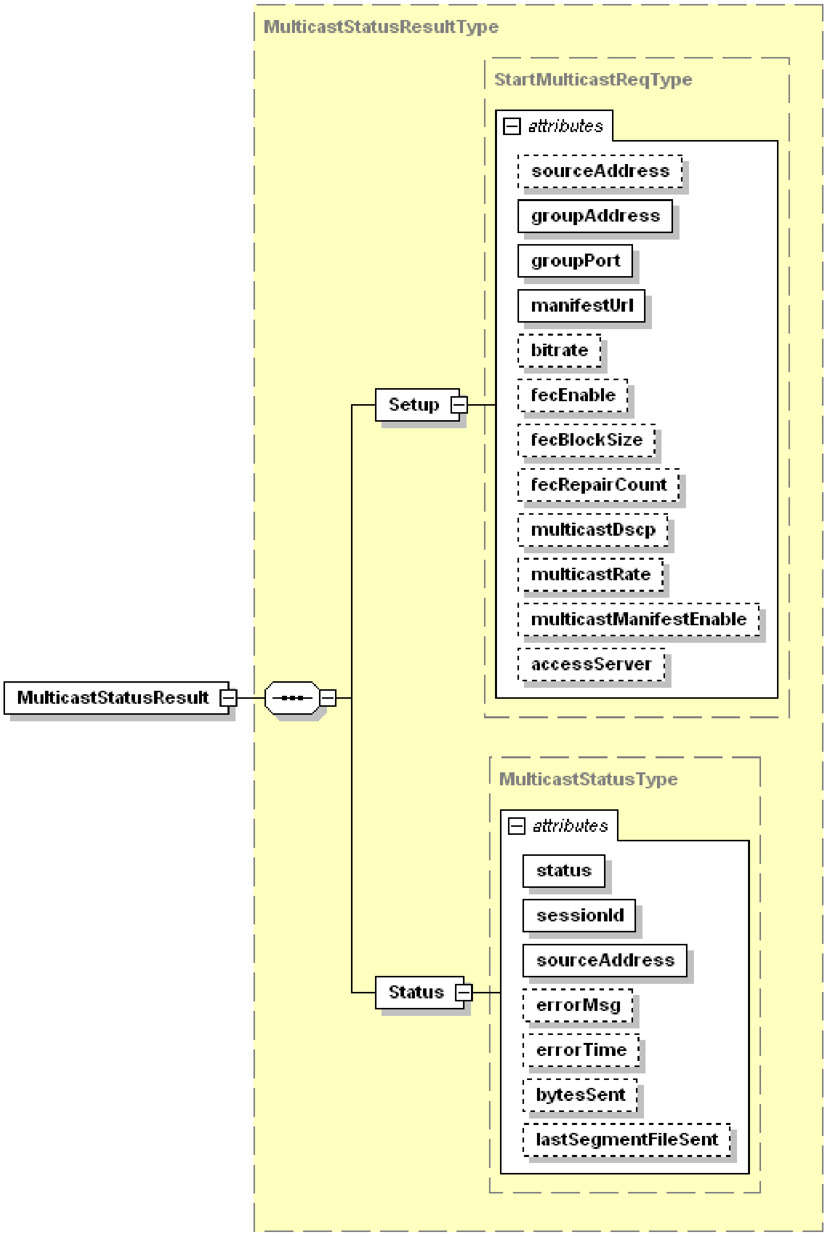

Figure 5 - MulticastStatusResult

Children:	Setup (1)
	Status(1)

7.2.2.1.1 Setup

The Setup element contains the same content as the original StartMulticastReq that initiated this stream. Refer to Table 1 for the specific attribute definitions.

7.2.2.1.2 Status

The Status element represents the current state of the session identified by its sessionId.

Children: None

Table 7 - Status Attribute Definitions

Attribute	Use	Data Type	Description
status	Required	xs:string	The current status of this session. This is a restriction with a base type of xs:string and the set of valid values are: “not-started" - the multicast was never started “running" - the multicast is actively running “stopped" - the multicast is stopped “error" - the multicast experienced a recoverable error “failed" - the multicast experienced an unrecoverable error
sessionId	Required	xs:string	A globally unique identifier for this multicast session.
sourceAddress	Required	xs:unsignedInt	The source address the Multicast Server utilized for this session.
errorMsg	Optional	xs:string	A descriptive error message. Always included when status is “error” or “failed”.
errorTime	Optional	xs:dateTime	The time an error most recently occurred. Always included when status is “error” or “failed”.
bytesSent	Optional	xs:unsignedLong	The number of content bytes which have been transmitted on this session.
lastSegmentFileSent	Optional	xs:string	The file name of the last segment file sent on this session.

7.2.2.2 MulticastStatusListResult

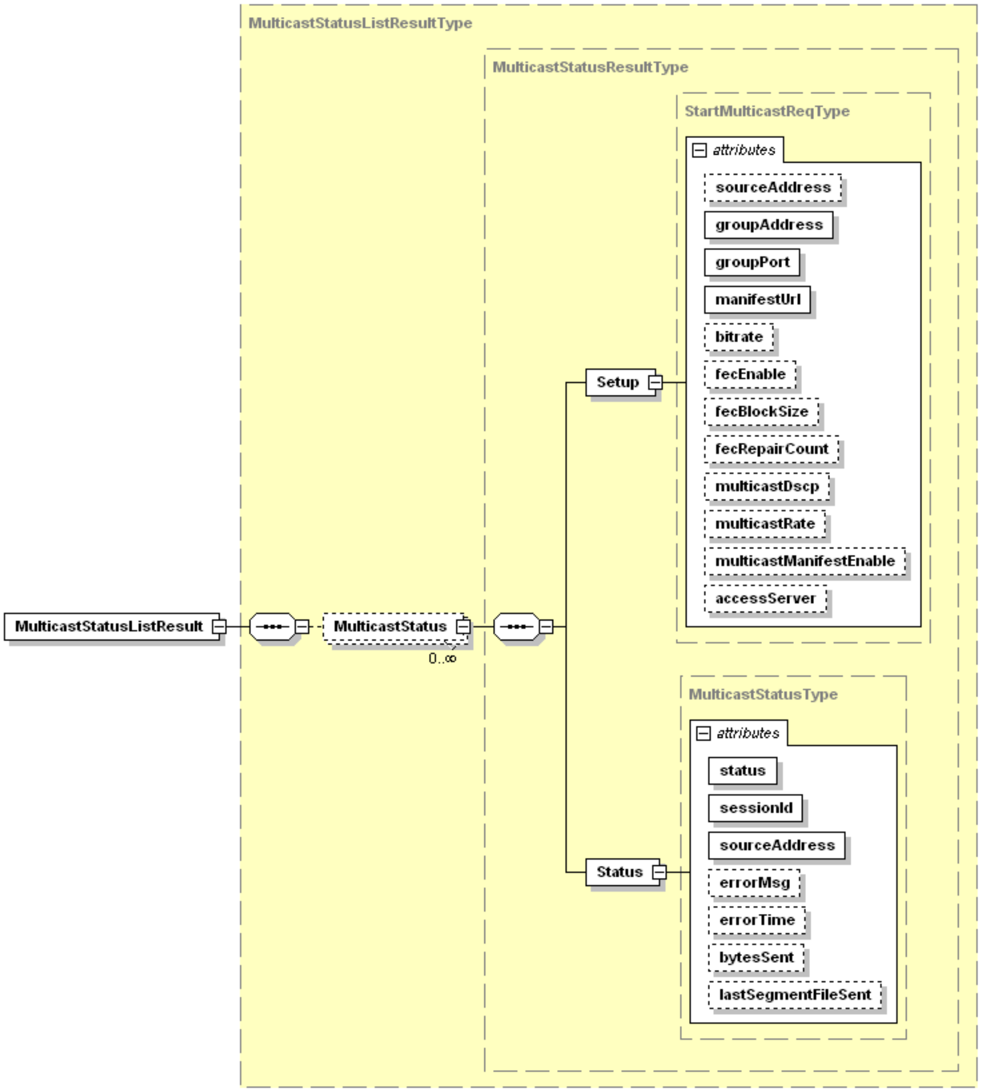

Figure 6 - MulticastStatusListResult

Children: MulticastStatus (0..n)

7.2.2.2.1 MulticastStatus

The MulticastStatus element contains a Setup and Status element as defined in Sections 7.2.2.1.1 and 7.2.2.1.2.

Children: Setup(1)

Status(1)

7.2.3 Multicast Status Examples

7.2.3.1 MulticastStatus Individual Request Example

```
GET /ms/multicast/a7cb6288-7b1d-11e4-b116-123b93f75cba HTTP/1.1
Host: ms
User-Agent: mc
```

7.2.3.2 MulticastStatus List Request Example

```
GET /ms/multicast/ HTTP/1.1
Host: ms
User-Agent: mc
```

7.2.3.3 MulticastStatusResult Response Example

```
HTTP/1.1 200 OK
Server: ms 1.1
Cache-Control: no-cache
<MulticastStatusResult>
  <Setup
    groupAddress="224.1.1.1"
    groupPort="12345"
    bit rate="3389000"
    manifestUrl="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_6713276793419826
163_HLS/AnEHD_HD_NAT_14710_0_6713276793419826163_HLS.m3u8"
  />
  <Status
    status="running"
    sessionId="a7cb6288-7b1d-11e4-b116-123b93f75cba "
    sourceAddress="10.10.10.123"
    lastSegmentFileSent="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_67132
76793419826163_HLS/format/hls/track/muxed/bandwidth/2946000/repid/01/duration
/180000/frag/435173/asset/20141114T175807-01-435173live.ts"
    bytesSent="4294628"
  />
</MulticastStatusResult>
```

7.2.3.4 MulticastStatusListResult Response Example

```
HTTP/1.1 200 OK
Server: ms 1.1
Cache-Control: no-cache
<MulticastStatusListResult>
 <MulticastStatus>
  <Setup
    groupAddress="224.1.1.1"
    groupPort="12345"
    bit rate="3389000"
    manifestUrl="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_6713276793419826
163_HLS/AnEHD_HD_NAT_14710_0_6713276793419826163_HLS.m3u8"
  />
  <Status
    status="running"
    sessionId="a7cb6288-7b1d-11e4-b116-123b93f75cba"
    sourceAddress="10.10.10.123"
    lastSegmentFileSent="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_67132
76793419826163_HLS/format/hls/track/muxed/bandwidth/2946000/repid/01/duration
/180000/frag/435173/asset/20141114T175807-01-435173live.ts"
    bytesSent="4294628"
  />
 </MulticastStatus>
 <MulticastStatus>
```

```
  <Setup
    groupAddress="224.1.1.1"
    groupPort="12346"
    bit rate="3389000"
    manifestUrl="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_6713276793419826
123_HLS/AnEHD_HD_NAT_14710_0_6713276793419826123_HLS.m3u8"
  />
  <Status
    status="error"
    sessionId="eb2ee266-7b2c-11e4-b116-123b93f75cba"
    sourceAddress="10.10.10.123"
    lastSegmentFileSent="http://10.12.12.12/hls/AnEHD_HD_NAT_14710_0_67132
76793419826123_HLS/format/hls/track/muxed/bandwidth/2946000/repid/01/duration
/180000/frag/435173/asset/20141201T175607-01-543173live.ts"
    bytesSent="65356"
    error=”404 File Not Found”
    errorTime=”2014-12-01T17:57:07.0Z”
  />
 </MulticastStatus>
</MulticastStatusListResult>
```

7.3 Multicast Channel Map Functions

7.3.1 Send Channel Map Request

To initiate transmission of a series of ABR segments to a given multicast group, the MC transmits a StartMulticast message to the MS.

Request Direction:	MC to MS
Method:	HTTP POST
Message URL:	`http://<device-url>/ms/SendChannelMap`
Children:	ChannelMap (1)

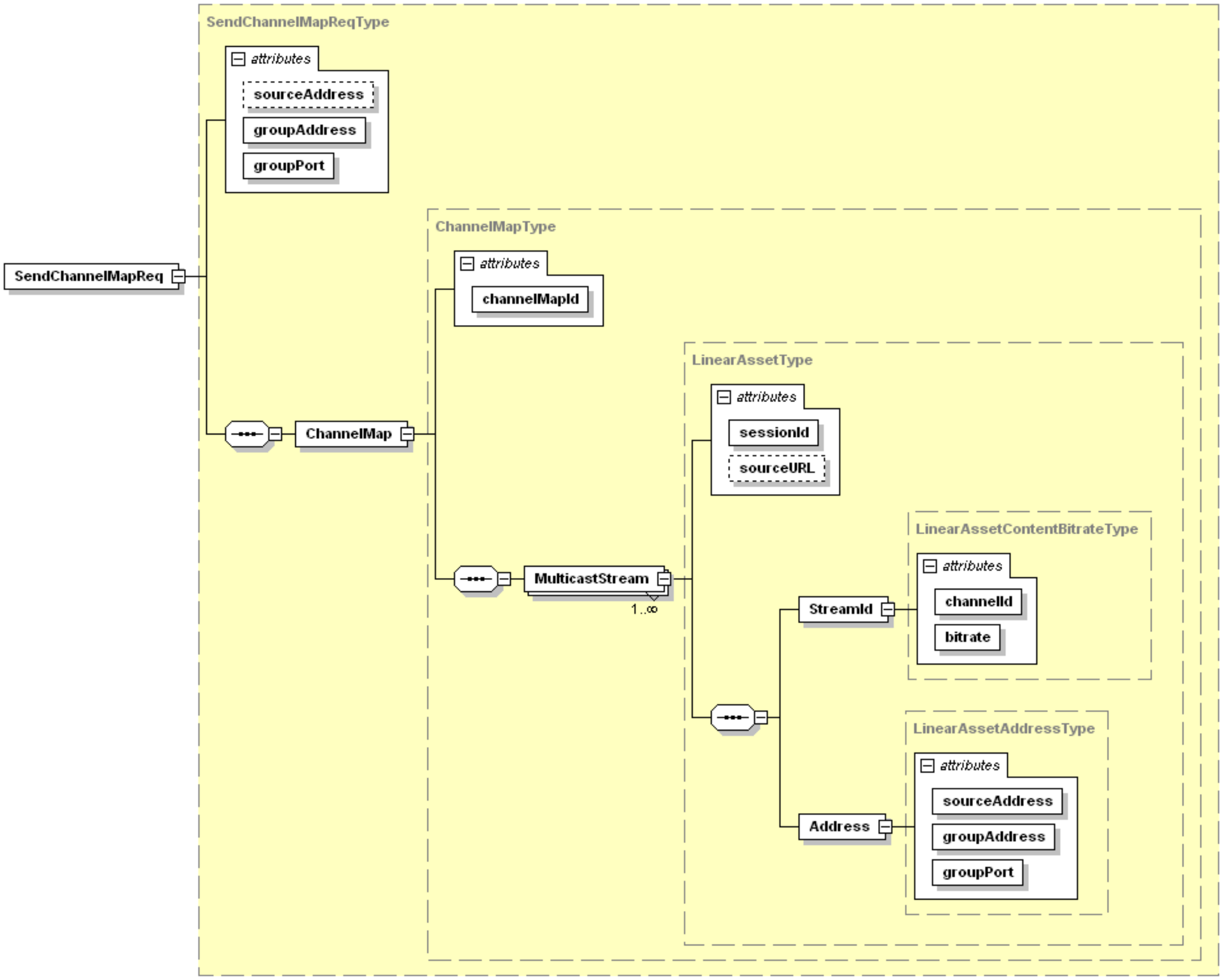

Figure 7 - SendChannelMapReqMsg

Table 8 - SendChannelMapReqMsg Attribute Definitions

Attribute	Use	Data Type	Description
sourceAddress	Optional	dvb:IPOrDomainType	The source address the MC instructs the MS to use for this (S,G). If omitted, the MS selects the sourceAddress.
groupAddress	Required	dvb:IPOrDomainType	The address the MC instructs the MS to use as the destination address for the multicast channel map transmission.
groupPort	Required	xs:short	The port the MC instructs the MS to use as the destination port for the multicast channel map transmission.

7.3.1.1.1 ChannelMap

The ChannelMap element is defined in [MC-EMC].

7.3.2 SendChannelMap Response

On successful completion of a SendChannelMap request, the Multicast Server MUST return 200 OK with an XML response carrying the SendChannelMapResult element, detailing the selected sessionId and other optional information. The SendChannelMapResult element is defined in Section 7.1.2.1.

On the failure of a SendChannelMap request on the MS, the Multicast Server MUST return 500 Internal Server Error and provide a SendChannelMapResult element with details about the failure in its Response element.

On a SendChannelMap request that leads to a standard HTTP protocol error, the Multicast Server MUST return the appropriate HTTP status code and status text.

Table 9 - SendChannelMap Response Status Codes

Status Code	Meaning
200	OK
400	Bad Request
500	Internal Server Error

7.3.2.1 SendChannelMapResult

The SendChannelMapResult element is returned with the XML response to a SendChannelMap request from the Multicast Controller when no HTTP errors occur. It provides the status of the response, details of any errors that occurred and the resulting impact of the request on the Multicast Server.

Children: Response(1)

StartMulticastDetails(0..1)

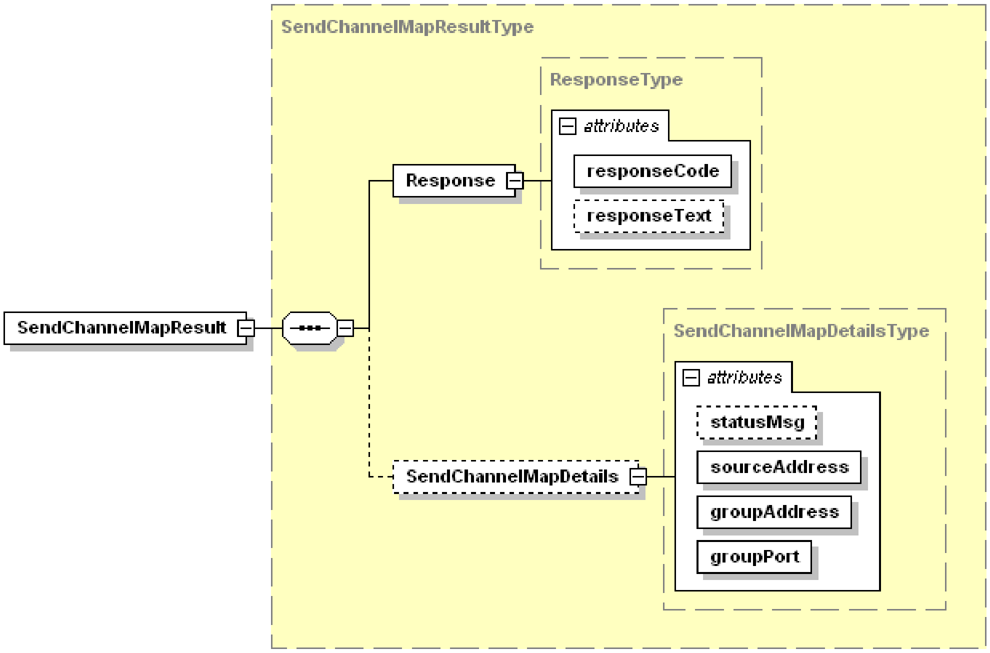

Figure 8 - SendChannelMapResult

7.3.2.1.1 Response

The Response element is defined in [MC-EMC]. The Multicast Server MUST provide a response code value of 200 on a successful completion of the SendChannelMap request. Standard SendMulticastRequest failure response codes are detailed in the Response Codes section of [MC-EMC].

7.3.2.1.2 StartMulticastDetails

Table 10 - StartMulticastDetails Attribute Definitions

Attribute	Use	Data Type	Description
statusMsg	Optional	xs:string	Any status message associated with this setup.
sourceAddress	Required	dvb:IPOrDomainType	The source address the MS was instructed or selected to use for this (S,G).
groupAddress	Required	dvb:IPOrDomainType	The group address the MS was instructed to use for this (S,G).
groupPort	Required	xs:short	The port the MS was instructed to use as the destination port for the multicast stream.

7.3.3 Send Channel Map Examples

7.3.3.1 SendChannelMap Request Example

```
POST /ms/SendChannelMap HTTP/1.1
Host: ms
User-Agent: mc
Content-Type: application/xml
Content-Length: ...
<SendChannelMapReq
    groupAddress="224.1.1.1"
    groupPort="12345"
    <ChannelMap>
      <MulticastStream
        sourceURL="https://www.tvcdn.net/espn2.m3u8?bit rate=3800&channelId=ESPNHD"
        sessionId="6c68ebc0-6ab0-11e4-b116-123b93f75cba">
        <StreamId
          channelId="ESPNHD"
          bit rate="3800000"/>
        <Address
          groupAddress="224.1.1.1"
          groupPort="12345"
          sourceAddress="10.10.10.10"/>
      </MulticastStream>
      <MulticastStream
        sourceUrl=" https://www.tvcdn.net/espn2.m3u8?bit rate=3800&channelId=ESPN2HD"
        sessionId="502faa5a-6a41-4384-8254-d4ff13f48f60">
        <StreamId
          channelId="ESPN2HD"
          bit rate="3800000"/>
        <Address
          groupAddress="224.1.1.2"
          groupPort="12345"
          sourceAddress="10.10.10.10"/>
      </MulticastStream>
    </ChannelMap>
/>
```

7.3.3.2 SendChannelMap Response Examples

7.3.3.2.1 SendChannelMap Response Example: Success

```
HTTP/1.1 200 OK
Server: ms 1.1
```

```
Cache-Control: no-cache
<SendChannelMapResult>
    <Response responseCode=”200”/>
    <StartMulticastDetails
        sourceAddress="10.10.10.123"
        groupAddress=”224.1.1.1”
        groupPort=”12345”/>
</SendChannelMapResult>
```

7.3.3.2.2 SendChannelMap Response Example: Server Error

```
HTTP/1.1 500 Internal Server Error
Server: ms 1.1
Cache-Control: no-cache
<SendChannelMapResult>
    <Response responseCode=“502”
        responseText=”Invalid sourceAddress”/>
</SendChannelMapResult>
```

Appendix I Acknowledgements

On behalf of the cable industry and our member companies, CableLabs would like to thank the following individuals for their contributions to the development of this specification.

Contributor	Company Affiliation
Dan Torbet	ARRIS
Sangeeta Ramakrishnan	Cisco
Coby Young	Comcast

Matt White – CableLabs

Andrew Sundelin – Consultant to CableLabs