Video
IP Multicast

IP Multicast Adaptive Bit Rate Architecture Technical Report

OC-TR-IP-MULTI-ARCH-C01-161026

CLOSED

Notice

This OpenCable™ technical report is the result of a cooperative effort undertaken at the direction of Cable Television Laboratories, Inc. for the benefit of the cable industry and its customers. You may download, copy, distribute, and reference the documents herein only for the purpose of developing products or services in accordance with such documents, and educational use. Except as granted by CableLabs in a separate written license agreement, no license is granted to modify the documents herein (except via the Engineering Change process), or to use, copy, modify or distribute the documents for any other purpose.

This document may contain references to other documents not owned or controlled by CableLabs. Use and understanding of this document may require access to such other documents. Designing, manufacturing, distributing, using, selling, or servicing products, or providing services, based on this document may require intellectual property licenses from third parties for technology referenced in this document. To the extent this document contains or refers to documents of third parties, you agree to abide by the terms of any licenses associated with such third party documents, including open source licenses, if any.

© Cable Television Laboratories, Inc. 2014-2016

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-TR-IP-MULTI-ARCH-C01-161026			
Document Title:	IP Multicast Adaptive Bit Rate Architecture Technical Report			
Revision History:	V01 - 11/12/14 C01 - 10/26/16			
Date:	October 26, 2016			
Status:	~~Work in Progress~~	~~Draft~~	~~Released~~	**Closed**
Distribution Restrictions:	~~Author Only~~	~~CL/Member~~	~~CL/ Member/ Vendor~~	**Public**

Trademarks

CableLabs® is a registered trademark of Cable Television Laboratories, Inc. Other CableLabs marks are listed at http://www.cablelabs.com/certqual/trademarks. All other marks are the property of their respective owners.

Contents

Figures

Tables

1 SCOPE

This technical report describes a reference architecture for multicast of Adaptive Bit Rate (ABR) video over an IP-based access network (sometimes referred to as Multicast-ABR, Multicast Assisted ABR or M-ABR). This technical report covers all major system components, the various functional groupings and the network interfaces necessary for delivery of services. The intended audience for this document includes developers of products utilizing IP multicast to deliver ABR video, and network architects who need to understand the overall IP multicast architecture framework.

This technical report describes the various facets of IP multicast. It contains the following information:

- A reference architecture;
- Description of the various functional groupings within the architecture;
- High level goals of the architecture;
- Detailed description of specific architectural components;
- Best practices and design tradeoffs when deploying M-ABR

1.1 IP Multicast Overview

This technical report defines a reference architecture for IP Multicast and a set of open interfaces that leverage popular and emerging communications technologies to support the rapid migration to IP-based video. It also describes both best practices and design tradeoffs that operators may encounter when deploying this service. Wherever practical, it also provides methods of quantifying the impact of alternative design decisions.

While the time-shifting behavior of many viewers has operators looking towards a future where video is delivered primarily over unicast IP, switched digital video (SDV) data shows that there is still a substantial portion of viewers who watch live linear television. Given the capacity requirements for delivering 100% unicast video and that current viewership data shows that much of that content would be redundant, operators are looking towards IP multicast to improve efficiency and minimize the capacity required for IP video transmission.

Over the past several years, operators have made substantial investments in "TV Everywhere" infrastructure for delivery of video to tablets, smart phones and other companion devices. The design of this infrastructure was driven largely by the capabilities of these companion devices and the technologies popular in that space. One of the key technologies used to support companion devices is HTTP-based adaptive bit rate (ABR) video streaming, which allows different devices with different screen sizes and different/changing network conditions to receive video content appropriate for their current environment.

In looking toward migrating to IP-based video delivery for primary screens, operators were faced with two basic choices - multicast delivery of IP video over RTP or unicast delivery of ABR video over HTTP. The first choice would have meant a largely separate and functionally redundant infrastructure for IP video delivery with substantial associated capital and operational costs. The second choice would have meant significant capacity upgrades for delivery of unicast video to all subscribers.

Recently, Multicast ABR (M-ABR) has emerged as a middle road between these two extremes. It allows operators to leverage the same/similar infrastructure used in their TV Everywhere deployments without having the capacity costs of 100% unicast video delivery.

The focus of this technical report is on:

- M-ABR - Segmented ABR video delivery such as that provided by HTTP Live Streaming (HLS), HTTP Dynamic Streaming (HDS), Microsoft Smooth Streaming (MSS) or MPEG-DASH. The focus is on emerging M-ABR approaches as opposed to multicast of CBR video over RTP, which is already well-documented in the literature.
- Live Linear TV - Once content has been time-shifted, the ability to multicast that content is greatly reduced, as no two users are likely to be at the same point in the stream at the same time.

- Full Video Feature Set - As a service focused on the primary screen, the expectation is that every feature supported by QAM-based video delivery is also supported by IP Multicast delivery. These include features such as ad insertion, emergency alert system (EAS), AMBER alert (AA), etc.
- DOCSIS Access Network - The emphasis is on delivery of this service via CCAPs over operators' DOCSIS access networks.

This technical report is a non-normative document. It describes a new application of many existing standards and protocols, but does not define any new protocols or interfaces. It is possible that follow-on work spurred by this technical report may produce normative documents if areas are identified which could benefit from standardization; however, this is not a goal of this document.

This technical report leverages other open standards and specifications wherever possible.

1.2 Terminology

This document uses three terms to call out specific approaches identified by the working group as worthy of consideration when designing an M-ABR service. These approaches fall into three categories: Best Practice, Tentative Best Practice and Design Considerations.

- ***Best Practices*** are techniques that the working group has identified as generally being the preferred design approach in a specific area.
- ***Tentative Best Practices*** are techniques that current data point to as being Best Practices (as defined previously); however, insufficient data exists to definitively call the practice a Best Practice.
- ***Design Considerations*** are critical tradeoffs and design decisions that operators need to make; however, different business models or other legitimate differences between operators mean that there is no universal Best Practice in this area.

2 INFORMATIVE REFERENCES

This technical report uses the following informative references.

[Eshet-14] "Multicast as a Mandatory Stepping Stone for an IP Video Service to the Big Screen," A. Eshet, J. Ulm, U. Cohen and C. Ansley, 2014, The Cable Show NCTA/SCTE Technical Sessions. http://www.nctatechnicalpapers.com/Paper/2014/2014-multicast-as-a-mandatory-stepping-stone-for-an-ip-video-service-to-the-big-screen

[Horrobin-13] "Pioneering IPTV in Cable Networks," J. Horrobin and G. Shah, October 2013, SCTE Cable-Tec Expo. http://www.scte.org/devams/cgi-bin/msascartlist.dll/ProductInfo?productcd=TS51

[IMC-08] "Watching Television Over an IP Network," M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, 2008, Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement. http://an.kaist.ac.kr/~sbmoon/paper/intl-conf/2008-imc-iptv.pdf

[MM TR] PacketCable Multimedia Architecture Framework Technical Report, PKT-TR-MM-ARCH-V02-051221, December 21, 2005, Cable Television Laboratories, Inc.

[Pantos-14] HTTP Live Streaming, draft-pantos-http-live-streaming-14, R.P. Pantos and W.M. May, October 14, 2014, Internet Engineering Task Force.

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

[RFC 5510] IETF RFC 5510, Reed-Solomon Forward Error Correction (FEC) Schemes, J. Lacan, V. Roca, J. Peltotalo, S. Peltotalo, April 2009.

[RFC 5740] IETF RFC 5740, NACK-Oriented Reliable Multicast (NORM) Transport Protocol, B. Adamson, C. Bormann, M. Handley, J. Macker, November 2009.

[Ulm-09] "IP Video Guide - Avoiding Pot Holes on the Cable IPTV Highway," J. Ulm and P. Maurer, October 2009, SCTE Cable-Tec Expo.

2.1 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) Secretariat, 46000 Center Oak Plaza, Sterling, VA 20166, Phone +1-571-434-3500, Fax +1-571-434-3535, http://www.ietf.org

3 TERMS AND DEFINITIONS

This document 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 - 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 document uses the following abbreviations:

ABR	Adaptive Bit Rate
BSS	Business Support System
CCAP	Converged Cable Access Platform
CER	Codeword Error Rate
CIR	Committed Information Rate
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
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
KPI	Key Performance Indicators
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

NMS	Network Management System
NORM	NACK-Oriented Reliable Multicast
PLR	Packet Loss Rate
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 IP MULTICAST OF VIDEO

5.1 Overview

The IP Multicast ABR architecture defined in this document describes a set of functional groups and logical entities, as well as a set of interfaces (called reference points) that support the information flows exchanged between entities.

This section provides:

- A statement on the intended scope of this document;
- An overview of the architecture, including a description of the main functional groupings (e.g., Home Network, Access Network, CDN, Multicast) and logical entities (e.g., CCAP, Gateway, Multicast Controller) within those groupings;
- A set of design goals for the IP Multicast architecture.

5.2 Scope

This document has three main goals:

1) Document the benefits of M-ABR delivery of live linear video compared to IP-based unicast retrieval. This includes examining serving group sizing and quantifying the efficiency gains in both the access and aggregation networks.
2) Document a reference architecture and
3) Identify and quantify design tradeoffs where multiple mechanisms exist to achieve the same goal.

The following topics are explicitly out of scope of this document. They may be briefly mentioned in this document as a comparison to the core topic, but they are not a focus of this document:

1) Streaming video over RTSP, RTMP, etc.
2) Unicast video delivery architectures.
3) Content other than video.
4) Multicast delivery over networks other than the DOCSIS access network (although Multicast assisted ABR will run over different access network technologies).

5.3 Overview

An overview of the IP Multicast architecture elements and functional groupings is illustrated in Figure 1.

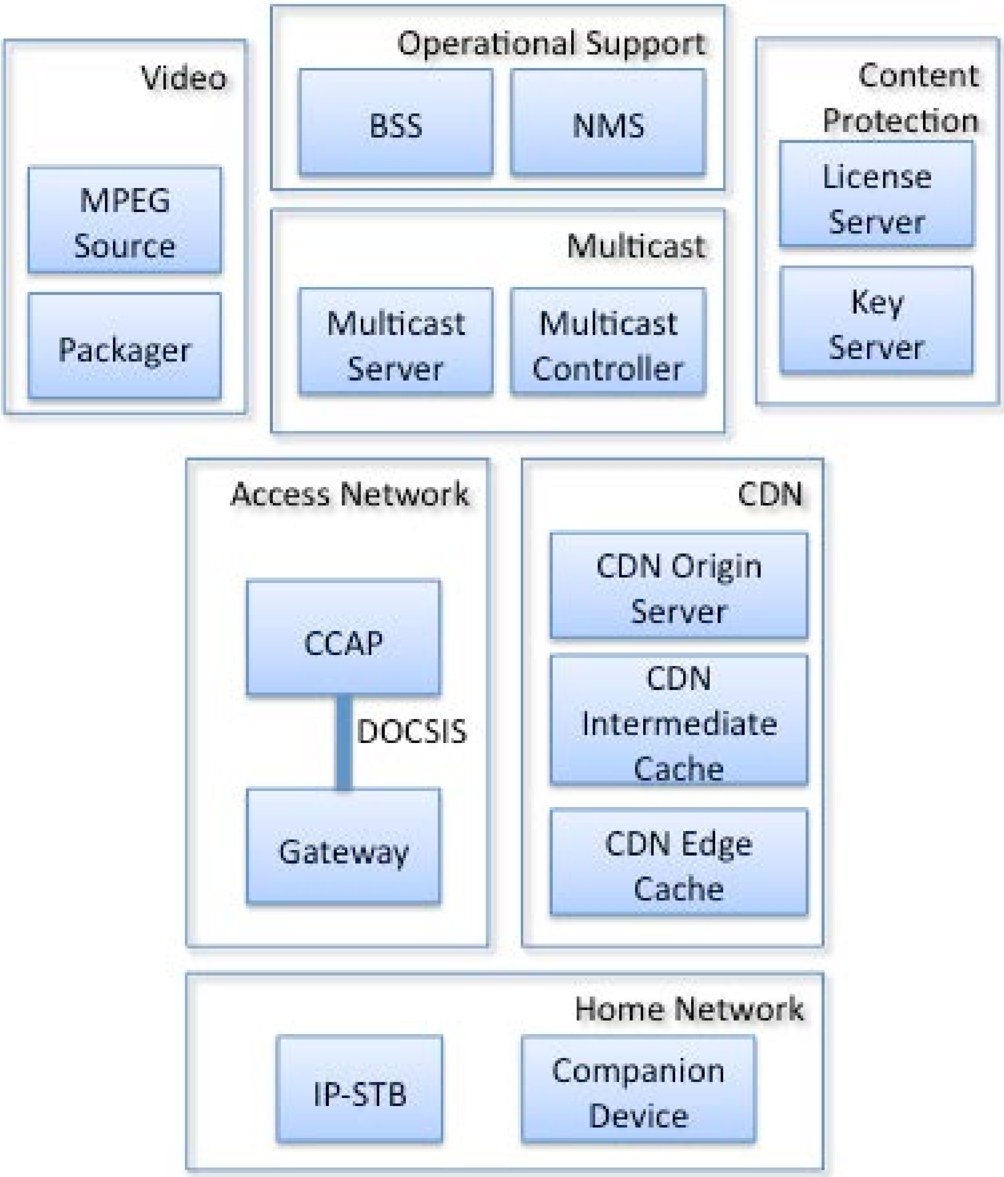

Figure 1 – M-ABR Elements and Functional Groupings

The elements are divided into several logical areas or functional groupings:

- Home Network: The home network is the network that the User Equipment (UE) uses to connect to the access network. It may be Ethernet, WiFi, MoCA or any other technology used to network or connect UEs. Home Network functional components include:
  - IP Set-Top Box (IP-STB): A set-top box which is only capable of receiving video over IP (i.e., it has no QAM demodulators). A Companion Device is customer owned and maintained (COAM) equipment that can be used to deliver IP video. Not shown in this figure, but equally important, is a software component referred to as the Player. The Player is any playback application that resides in the IP-STB or Companion

Device and is capable of playing ABR video. In this technical report the term Player is often generically used to refer to a playback entity which could reside on either the IP-STB or Companion Device.

- Gateway (sometimes referred to as the Residential Gateway): A CPE that contains an embedded DOCSIS cable modem (CM) for communicating to the CCAP and as a data path to the home network.
- Hybrid STB (not shown): A set-top box capable of terminating both QAM video and DOCSIS. This device has the capability to function as a Gateway.

- Access Network: The network between the customer's Gateway and the operator's edge network.
- CDN: The CDN often consists of three components - the origin server, midtier or shield cache and edge cache. Gateways direct requests for unicast video content to their edge cache (often with the assistance of a CDN selector). If the content is not present in the edge cache, the edge cache requests the content from the midtier cache. Similarly, if the midtier cache does not contain the content, it requests it from the origin server.
- Video or Source Video: The originating MPEG Source for the ABR-encoded video. The Packager segments the video stream and, typically, applies encryption.
- Content Protection: The License Server generates licenses for client devices derived from the keys it receives from the Key Server. The Key Server generates keys and supplies them to the encrypting device (i.e., the Packager). The Certificate Server manages identities for clients that will decrypt content.
- Operational Support: the Network Management System (NMS) monitors the other components involved in video delivery.

Note that some of the functional components described above are logical functions, which may be combined on common platforms.

5.4 IP Multicast Design Goals

In order to enable M-ABR across the cable network infrastructure, the IP Multicast reference architecture was designed to meet goals in a number of functional areas:

- Media Stream Transport & Delivery;
- Quality of Service;
- Viewership Measurement;
- Security.

5.4.1 Generic Architecture Goals

The design goals of the IP Multicast reference architecture include:

- Provide an architecture that transparently supports existing standard ABR players without modification (no additional client required);
- Provide an architecture that supports IP multicast delivery of ABR video over the access network to a Gateway;
- Provide an architecture that supports seamlessly switching between content delivered via unicast or multicast;
- Provide an architecture that supports switching between ABR video streams (multicast or unicast) with no additional delay when compared to changing channels between two unicast ABR streams and on the order of the channel change time for QAM-based video;
- Provide an architecture that supports all of the features of legacy QAM-based video, including Emergency Alert (EAS, EAN, etc.), Amber Alert, ad insertion (both by ad zone and subscriber);
- Provide a modular architecture, where architectural components can be combined in a variety of ways to support a wide range of features;

- Provide facility to provide dynamic multicast management for higher access network efficiencies;
- Provide a unicast failback mechanism when multicast is unavailable;
- Provide a means of reliable multicast transport;
- Support IPv4 and IPv6 operation;
- Leverage existing standards and open protocols whenever possible;
- No use of proprietary or patented technologies that would require licensing fees.

5.4.2 IP Version Support and Interworking

The design goals for IP Version support and interworking include:

- Support IPv4 and IPv6 for Gateways, IP-STBs and Companion Devices;
- Support network components in the CDN, Video, Content Protection, Operations Support, and Multicast functional groupings that operate in the following modes: IPv6-only, or IPv4-only, or IPv6/IPv4;
- Support IPv6 Gateways using MLDv2 accessing content sourced from an IPv4 multicast network.

5.4.3 Media Stream Transport & Delivery

Media Stream Transport & Delivery design goals include:

- Support for multicast transport protocol which allows endpoints to identify missing content and retrieve that content in unicast or resend in multicast;
- Support for multicast transport protocol with configurable transport layer FEC.

5.4.4 Quality of Service

QoS design goals include:

- Minimize the potential for unicast retransmissions (and, thus, wasted bandwidth) by minimizing latency and packet loss in M-ABR streams;
- Optional support for admission control so operators can ensure sufficient video capacity is available in their multi-service access networks;
- Support packet marking and classification from the access network such that a QoS mechanism like Differentiated Services (DiffServ) can be used in the backbone and edge network.

5.4.5 Viewership Measurement

Viewership Measurement design goals include:

- Enable the ability to periodically account for complete viewership;
- Allow for multiple network elements to generate events that can be correlated to a given session or subscriber;
- Support the correlation of accounting events across the signaling and bearer planes;
- Provide an OSS interface on the multicast controller that can report statistics from the multicast clients and multicast server.

5.4.6 Security

Security design goals include:

- Support for confidentiality, authentication, integrity, and access control mechanisms;
- Support for any Content Protection or Digital Rights Management system;
- Protection of the network from various denial of service, network disruption, theft-of-service attacks;

- Protection of the gateway from denial of service attacks, security vulnerabilities, unauthorized access (from network).

6 IP MULTICAST FUNCTIONAL COMPONENTS

This section provides additional detail on each of the functions in the Multicast-ABR architecture.

6.1 Home Network

6.1.1 IP Set-Top Box (IP-STB)

The IP-STB is the interface between the IP network and large screen displays. Its functions include:

- Identify content;
- Request content;
  - Request manifest and content;
  - Adapt content requests to observed network conditions;
- Ingest content;
- Display/playback content.

6.1.2 Companion Devices and Consumer Electronics

Consumer Electronics (CE) devices such as Smart TVs and media players as well as Companion Devices (both of which are referred to as Customer Owned And Managed (COAM) Devices) such as tablets, PCs, or smart phones. These devices have some functions similar to those of an IP-STB; they include:

- Request content;
  - Request manifests and content;
  - Adapt content requests to observed network conditions;
- Receive content;
- Manage security (certificates, keys);
- Decrypt content;
- Display/playback content.

This technical report uses the generic term Player to encompass both IP-STBs and Companion Devices.

6.2 Access Network

The Home Network connects to the Content Delivery Network via the existing cable access network. The Access Network elements provide the IP connectivity and QoS resources needed by the Home Network to provide video services as well as the physical and MAC layers in the OSI model.

6.2.1 Gateway (GW)

The Gateway provides the physical and logical interfaces between Multicast ABR sourced from the network and standard ABR video delivery to the requesting client. It is an active participant in multicast and unicast ABR video delivery. Its functions include:

- Caching Proxy:
  - Functions as a transparent proxy[1] for HTTP requests for segmented video content;
  - Caches content segments to reduce latency;

[1] Unlike standard proxies that require end-point configuration to utilize, transparent proxies require no end-point configuration. They operate by detecting HTTP traffic and requesting content on behalf of end-points in ways intended to improve performance.

- Requests unicast content from the CDN when content not found in its local cache;
- Ages out cached content segments when it is determined that they are no longer needed.

- Multicast:
  - Uses a multicast channel map to identify content available via IP multicast;
  - Receives periodic updates of multicast channel map and analyzes map to see if any previously unicast streams are available in multicast;
  - Joins multicast groups to acquire requested content available via multicast;
  - Fills cache with content acquired via multicast in advance of unicast requests from IP-STBs or COAM equipment;
  - Uses multiple multicast caching buffers to improve multicast efficiency.
- Signaling:
  - Signals the Multicast Controller of desired content streams when those streams are not currently available via multicast;
  - Constantly updates multicast controller at prescribed intervals with what content is being consumed in the household;
- Manifest Manipulation:
  - Manipulates manifest files for performance improvements (e.g., trims references to number of segments).[2]

6.2.2 Converged Cable Access Platform (CCAP)

The CCAP[3] controls multicast transmission and QoS on the access network. In this role it optimizes the mapping of multicast content to bonding groups as well as the assignment of multicast listeners to bonding groups. It also enables downstream QoS for multicast as well as authorizing group membership requests. Its functions include:

- Multicast Control:
  - Optimize multicast replication across DOCSIS bonding groups within a serving group;
  - Manage bonding groups assigned to CMs for multicast efficiency (a.k.a. multicast-aware load-balancing);
  - Allow or deny multicast Joins from clients based on configured rules (DOCSIS Multicast Join Authorization);
  - Optimize assignment of DOCSIS 3.1 OFDM profiles for multicast receiver set;
  - IGMP/MLD translation proxy for clients;
- QoS:
  - Admission control;
  - Group Service Flows and Classifiers to enable QoS on multicast sessions via a configured set of rules;
  - Multicast Service Flows and Classifiers with associated QoS established via PacketCable Multimedia (PCMM). (This is an alternative to using the Group Service Flow approach to enable multicast QoS.)
- Reporting:
  - Multicast usage monitoring.

[2] In order to facility delivery of multicast segments to the Gateway before the same segments are advertised to the Player, the manifest is shortened such that the Gateway is aware of more segments than the Player. This is to avoid a race condition where a Player requests a segment before it is available via the Gateway cache. Note: Manifest manipulation can also be performed within the infrastructure to simplify the Gateway.

[3] This technical report refers to the DOCSIS aggregation and control element as a CCAP, but this could also be a CMTS.

6.3 Content Delivery Network (CDN)

The Content Delivery Network varies from operator to operator, but for reference this technical report considers the following components as part of the CDN function.

6.3.1 CDN Edge Cache

The CDN Edge Cache functions include:

- Providing content at the edge of the network where the latency is lowest;
- Accessing content from other caches within the hierarchy of the CDN.

6.3.2 CDN Midtier or Shield Cache

The CDN Midtier or Shield Cache functions include:

- Acting as an intermediary between the origin cache/server and the edge cache;
- Providing additional caching intended to minimize traffic, reduce load on origin servers, provide redundancy and/or minimize latency.

6.3.3 CDN Origin Server

The CDN Origin Server functions include:

- Receiving and storing video content from the Packager or Transcoder;
- Serving video content to subordinate caches in the CDN hierarchy.

6.4 Multicast Components

These components are key to the Multicast function of the Multicast-ABR reference architecture.

6.4.1 Multicast Controller

The Multicast Controller controls the availability of content in the multicast streams that the Multicast Server injects into the network. It also decides how to map content to multicast groups and, thus, controls the channel map. Its functions include:

- Content Selection:
    - Policy Driven - BSS pushes policy rules to the Multicast Controller via its provisioning interface. The Controller then regulates content/bit rates to be carried in multicast via the control interface on the Multicast Server.
    - Viewership Driven - Multicast Controller autonomously determines content/bit rates to be carried in multicast based on client requests for content and available bandwidth for multicast streams.
- Control Multicast Delivery:
    - Interface to the Multicast Server to control the content/bit rates that the MS will transmit via multicast.
- Managing/Delivering Channel Maps:
    - Inform channel-mapping service of current multicast channel map and/or deliver channel map to multicast clients.
- Reporting:
    - Export data on multicast content/bit rates made available to OSS.

6.4.2 Multicast Server

The multicast server acquires and transmits content via multicast as directed by the Multicast Controller. Its functions include:

- Ingesting:
  - Acquire content from just-in-time Packager or the CDN Origin.
- Processing:
  - Insert metadata (e.g., HTTP headers associated with the original unicast version of a segment);
  - Efficiently pack and encapsulate content into transport protocol.
- Streaming:
  - Disseminate multicast address for content/bit rate combination;
  - Transmit packets into appropriate multicast group;
  - Control output packet rate;
  - Mark packets with appropriate DSCP value.

6.5 Video

These components are the core of the Video function of the Multicast-ABR reference architecture.

6.5.1 MPEG Source

The MPEG Source functions include:

- Streams live linear MPEG video content to Packager.

6.5.2 Packager

The Packager functions include:

- Ingests live linear video streams from MPEG source;
- Segments video stream into fixed duration files;
- Encrypts video segments;
- Delivers segmented video files to Multicast Server and/or CDN Origin.

6.6 Operational Support Systems

These components are key to the Operational Support Systems function of the Multicast-ABR reference architecture.

6.6.1 Business Support System (BSS)

The BSS functions include:

  - Policy-based control of content/bit rates to be multicast (optional).

6.6.2 Network Management System (NMS)

The NMS functions include:

  - Viewership monitoring;
  - Multicast stream monitoring.

6.7 Content Protection

Content Protection for Multicast ABR is not covered in detail in this technical report as it is identical to the function of the DRM-based Content Protection system for unicast ABR. However, for completeness some of the basic functions of these components are included here.

6.7.1 License Server

The License Server functions include:

- Provides licenses to IP-STB and Companion Devices;
- Interfaces to the Key Server to populate licenses with keys appropriate for the requested content.

6.7.2 Key Server

The Key Server functions include:

- Provides content encryption keys to Packagers;
- Provides decryption keys to License Servers.

6.7.3 Certificate Server

The Certificate Server functions include:

- Provides individualization certificate to each client (IP-STB or COAM device).

7 REFERENCE INTERFACES

This technical report documents a set of protocol interfaces between many of the functional components in the reference architecture. These interfaces are defined solely to inform and facilitate discussion around the overall system and should not be interpreted as normative.

It is possible that some of these reference points may not exist in a given operator's or vendor's implementation. For example, if several functional components are integrated, then it is possible that some of these reference points are internal to the integrated device.

7.1 Reference Interface Definition

The Multicast ABR interfaces defined in this report are illustrated in Figure 2.

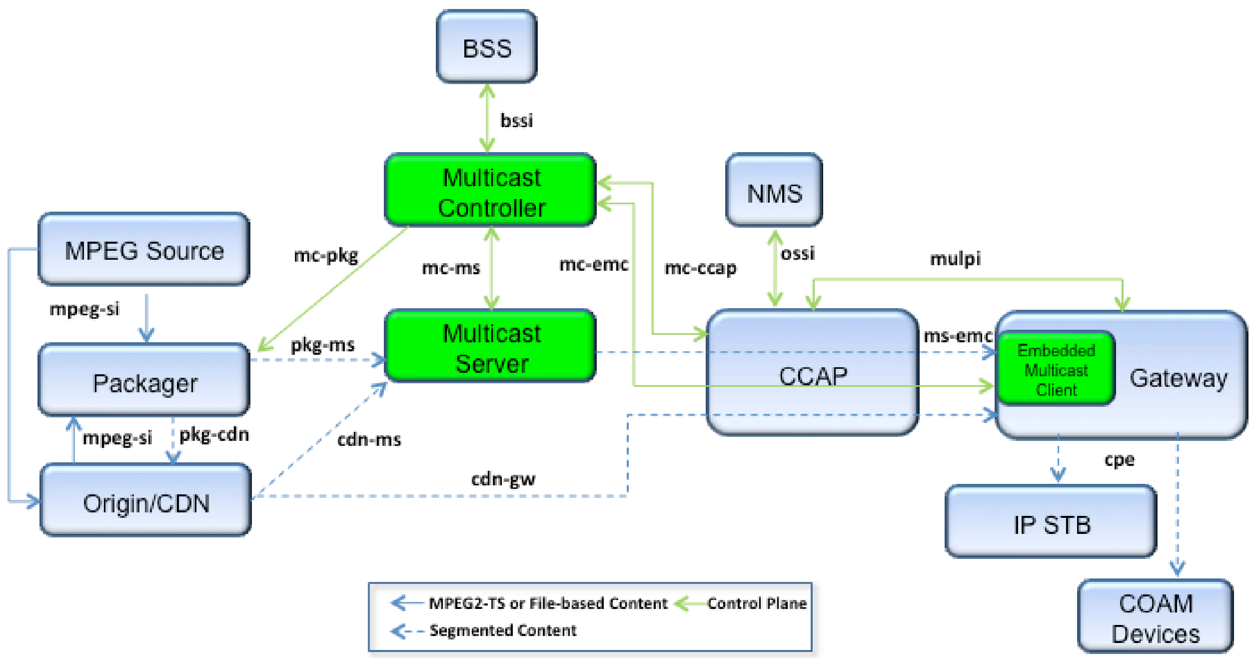

Figure 2 – Multicast ABR Reference Interfaces

The reference points depicted in Figure 2 are described in Table 1.

Table 1 – Multicast ABR Reference Interface Descriptions

Interface	Reference Point Description
cpe	Allows a player on an IP-STB or COAM device to view standard ABR video. While multiple protocols can traverse this interface, they are all completely standard web/Internet protocols - the multicast aspect of the system is completely transparent to these devices and their players.
ms-emc	Allows the Multicast Server to multicast video segments to the Embedded Multicast Client within the Gateway. The Gateway's embedded transparent proxy then caches the video segments for retrieval via the cpe interface.
mc-emc	Allows the Gateway's Embedded Multicast Client to identify the multicast group of video content available on multicast.

Interface	Reference Point Description
cdn-gw	Allows for unicast content retrieval of content which is unavailable via multicast. Like the cpe interface, this interface is completely based on standard web/Internet protocols for ABR video retrieval.
mulpi	Allows for QoS-enabled delivery of multicast content over the access network.
mc-ccap	Allows for Multicast Controller to perform multicast QoS discovery or other CCAP synchronization.
mc-ms	Allows the Multicast Controller to determine the content and bit rates the Multicast Server is delivering via multicast.
cdn-ms	Allows the Origin Server/CDN to provide the Multicast Server with video segments ready for multicast delivery.
pkg-ms	Allows the Packager to provide the Multicast Server with video segments ready for multicast delivery.
pkg-cdn	Allows the Packager to provide video segments to the CDN for unicast retrieval.
mc-pkg	Allows the Multicast Controller to determine the content and bit rates the Packager is providing to the Multicast Server.
mpeg-si	Allows the MPEG Source to provide video source material directly to the Packager or indirectly to the Packager via the Origin Server/CDN.
bssi	Allows the Business Support System to provide policy to the Multicast Controller for determining the content and bit rates to be multicast.
ossi	Allows the CCAP to report multicast viewership to the NMS.

7.1.1 CPE Interface

The specifics of this interface are largely out of the scope of this technical report as they are standard mechanisms for unicast ABR video retrieval. While critical for actual delivery of content to viewers, it is a requirement that Multicast-ABR not require any changes to this interface - any M-ABR delivery is required to be entirely transparent to this interface.

Purpose:	Standard unicast retrieval of ABR video content by video clients.
Control Protocol:	HTTP Live Streaming (HLS), HTTP Dynamic Streaming (HDS), HTTP Smooth Streaming (HSS), MPEG-DASH
Content Container:	Varies

7.1.2 Multicast Server - Embedded Multicast Client (ms-emc) Interface

This interface provides the actual multicast transport of video segments from the Multicast Server to the Embedded Multicast Client.

Purpose:	Deliver ABR video segments to the Embedded Multicast Client within the Gateway.
Control	IGMPv3/MLDv2 (membership)

Protocol:	NORM (transport)
Content Container:	ISO base media file format or MPEG-2 Transport Stream

7.1.3 Multicast Controller - Embedded Multicast Client (mc-emc) Interface

As the Gateway that both IP-STBs and Companion Devices utilize to access content is under operator control, this interface is operator-specific. Activity on this interface is generally triggered by activity on the cpe interface.

Purpose:	Deliver channel mapping data (content to (S,G) mapping) to client Provide channel change data to Multicast Controller
Control Protocol:	Operator specific (HTTP, NORM (channel map), etc.)
Content Container	Operator specific (HTML, XML, JSON, etc.)

Some operators unicast their channel map data and other operators are looking to multicast the channel map data. Section 9.2.3 discusses channel map design considerations in some detail.

7.1.4 CDN - Gateway (cdn-gw) Interface

Like the cpe interface, the specifics of this interface are largely out of scope of this technical report as they are standard mechanisms for unicast ABR video retrieval via transparent proxy. This interface is largely for the retrieval of ABR video content which is not available via multicast.

Purpose:	Standard unicast retrieval of ABR video content by the Gateway.
Control Protocol:	HTTP Live Streaming (HLS), HTTP Dynamic Streaming (HDS), HTTP Smooth Streaming (HSS), MPEG-DASH
Content Container:	Varies

7.1.5 MULPI Interface

This interface provides access network QoS for multicast delivery of video segments.

Purpose:	DOCSIS delivery of multicast video segments from the CCAP to the Gateway.
Control Protocol:	DOCSIS MAC; IGMPv3/MLDv2

7.1.6 Multicast Controller - Multicast Server (mc-ms) Interface

This interface is internal to operators and is largely either vendor or operator proprietary.

Purpose:	Control the content and bit rates being delivered to the Embedded Multicast Client via multicast.
Control Protocol:	Operator specific (REST, etc.)

Content Container:	Operator specific (XML, JSON, etc.)

7.1.7 Packager - Multicast Server (pkg-ms) Interface

This interface is internal to operators and is largely either vendor or operator proprietary.

Purpose:	Provide encrypted video segments for delivery via multicast
Control Protocol:	HTTP
Content Container:	MPEG-2 Transport Stream files

7.1.8 CDN - Multicast Server (cdn-ms) Interface

This interface is internal to operators and is largely either vendor or operator proprietary.

Purpose:	Provide encrypted video segments for delivery via multicast
Control Protocol:	HTTP
Content Container:	MPEG-2 Transport Stream files

Typically, either the cdn-ms or the pkg-ms interface, but not both, is instantiated as part of an M-ABR deployment. Said slightly differently, some operators have their Multicast Servers pull directly from their Packager while others have their Multicast Servers pull from their CDN. In either case, the interface instantiated on the Multicast Server is largely the same - it generally acts as an HTTP client and pulls content from the Packager or the CDN.

7.1.9 Multicast Controller - Packager (mc-pkg) Interface

This optional interface is internal to operators and is largely either vendor or operator proprietary.

Purpose:	Control the content and bit rates of video segments being generated by the Packager for the Multicast Server.
Control Protocol:	Operator specific (REST, etc.)
Content Container:	Operator specific (XML, JSON, etc.)

Note: this interface is optional. In some architectures the set of content being packaged (including bit rates, etc.) is independent of the multicast service. In this case, the Multicast Server is told the subset of the overall set of packaged content to multicast, but the Packager may not even be aware of the multicast-related systems.

7.1.10 MPEG Source Interface (mpeg-si)

This interface is the start of the processing chain for Multicast-ABR where live linear content is streamed for real-time packaging and subsequent multicast delivery.

Purpose:	Provide live linear MPEG-TS streams directly or indirectly (via Origin Server) to the Packager for segmentation and encryption.
Control Protocol:	Out of scope
Content Container:	MPEG-2 Transport Stream

7.1.11 BSS Interface (bssi)

This interface provides an optional mechanism for the BSS to inject policy into the Multicast Controller's decisions about what content/bit rates to provide via multicast.

Purpose:	Control the content/bit rates the Multicast Controller tells the Packager and Multicast Server to prepare for multicast delivery.
Control Protocol:	Operator specific (REST, etc.)
Content Container:	Operator specific (XML, JSON, etc.)

7.1.12 OSS Interface (ossi)

This interface provides a mechanism for the CCAP to provide data on multicast usage and per gateway multicast viewership data.

Purpose:	Primarily, multicast usage & viewership reporting.
Control Protocol:	IPDR/SP; SNMP
Content Container:	XDR; BER

7.2 Security

Content Protection for Multicast ABR is not covered in detail in this technical report, as it is identical to the function of the Content Protection system for unicast ABR.

7.3 Functional Overview

A typical M-ABR system can be thought of as a standard ABR video system, which uses a transparent caching proxy resident in the Gateway. That transparent cache can be filled either via unicast or multicast. This allows the Player to switch seamlessly between less-popular content only available on unicast and popular content available on multicast as it is completely transparent to the Player whether the content is delivered to the Gateway via unicast or multicast. In fact, the system can switch seamlessly between unicast and multicast delivery of the same stream, as any content not delivered by multicast will be retrieved via unicast.

While this technology is referred to as "Multicast Adaptive Bit Rate (M-ABR)", it is important to note that individual multicast streams do not "adapt" their bit rates. Rather, the term is used to refer to the multicast delivery of video segment files to the Gateway which subsequently delivers these segments via HTTP when they are requested by a streaming video Player. Each multicast stream only contains a single bit rate. The pre-filling of the

Gateway's cache is expected to result in the reliable receipt of fragments by the Player, such that the Player does not adapt and instead chooses to remain at that bit rate. However, for robustness, the manifest still generally contains reference to other bit rate encodings of the same content stream. These other bit rates can be provided on a separate multicast stream or may be available only via unicast retrieval.

7.3.1 Basic Player Retrieval

The basic model for the retrieval of new content by a Player is shown in the following figure:

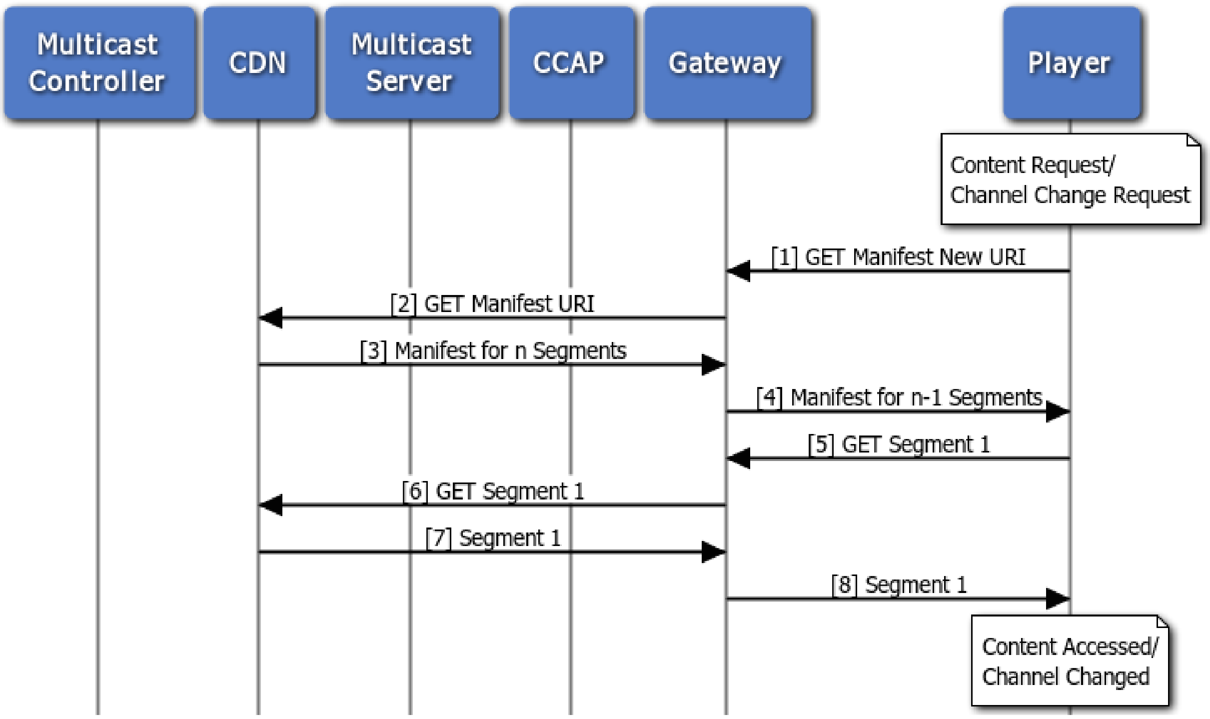

Figure 3 – Initial Retrieval of New Content

It is important to note that there are no multicast-related steps in this sequence diagram. This sequence is identical to the sequence which would occur in a unicast system with a transparent caching proxy - with one very small exception: between Step 3 and Step 4 the Gateway modifies the manifest by dropping the last segment from the list.[4] This way the Gateway always knows about one more segment than the Player is aware of. While not important for this initial content delivery or a channel change, this is important feature for multicast delivery and will be explained in more detail later.

It is also important to note that since the steps taken are identical to that for unicast, the performance of an M-ABR system is never worse than that of unicast and thus the QoE for the end user is also never worse than normal ABR retrieval. This applies to both initial content access and channel changes.

Figure 3 only shows the retrieval of the very first video segment file in the content stream. The following figure takes this one step further by showing two new aspects of this system:

1) The retrieval of multiple manifests and multiple segments as a video is watched.
2) The Gateway checking to see if a video segment is available in the cache before fetching the segment from the CDN.

Except for the fact that the Gateway's cache can be pre-filled by multicast delivery, the system in the figure behaves just like a unicast transparent caching proxy would.

[4] Shortening the manifest which gets delivered to the Player can also be performed in the back office. Some operators have two versions of the manifest for a given content stream – a shortened one for Players and a longer version for other components.

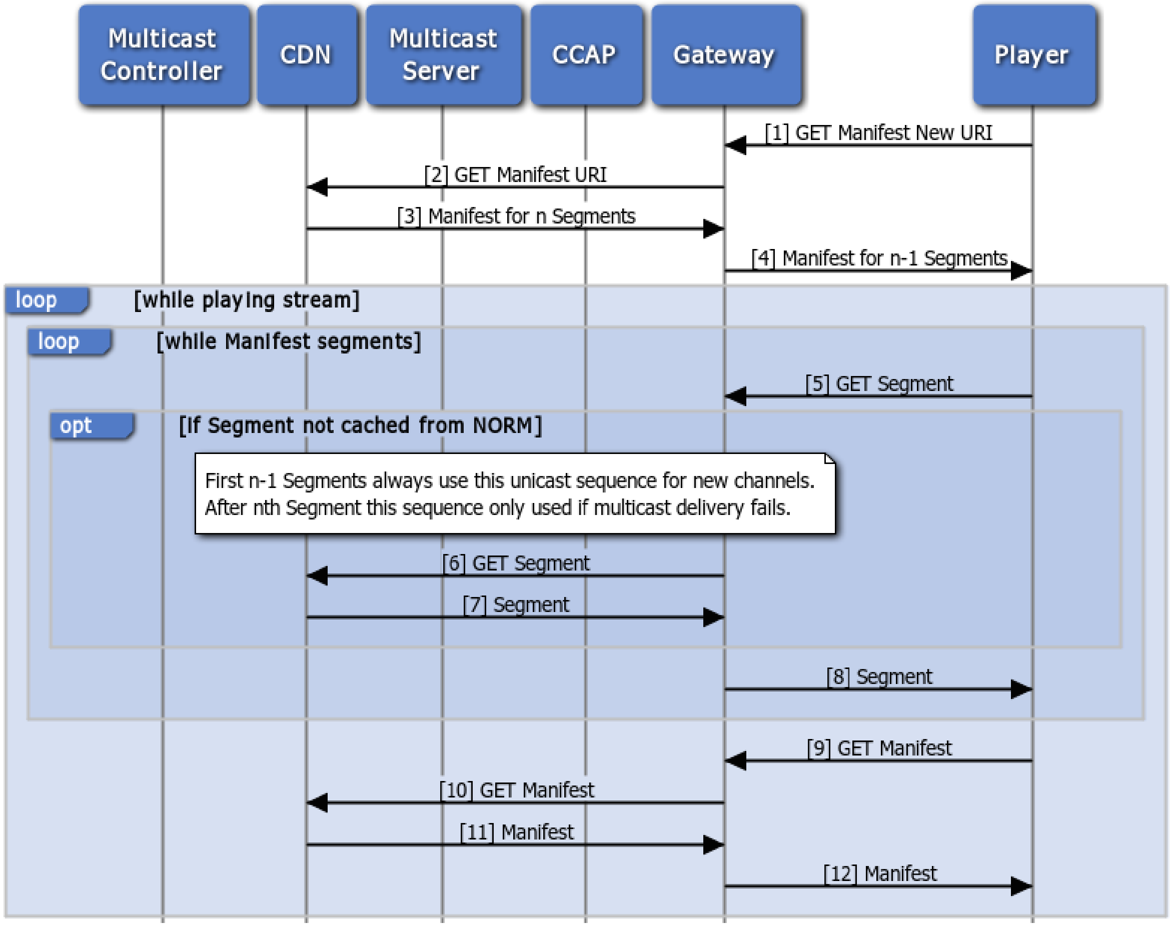

Figure 4 – Continuous Delivery of New Content

Thus, playback and channel change functions are virtually identical to unicast. Similarly, the performance in terms of channel change times and other QoE metrics is identical to unicast. Where the system differs from unicast is in the way the cache can be filled if a given content stream is available via multicast.

The following figure shows how the caching subsystem utilizes multicast to fill its cache.

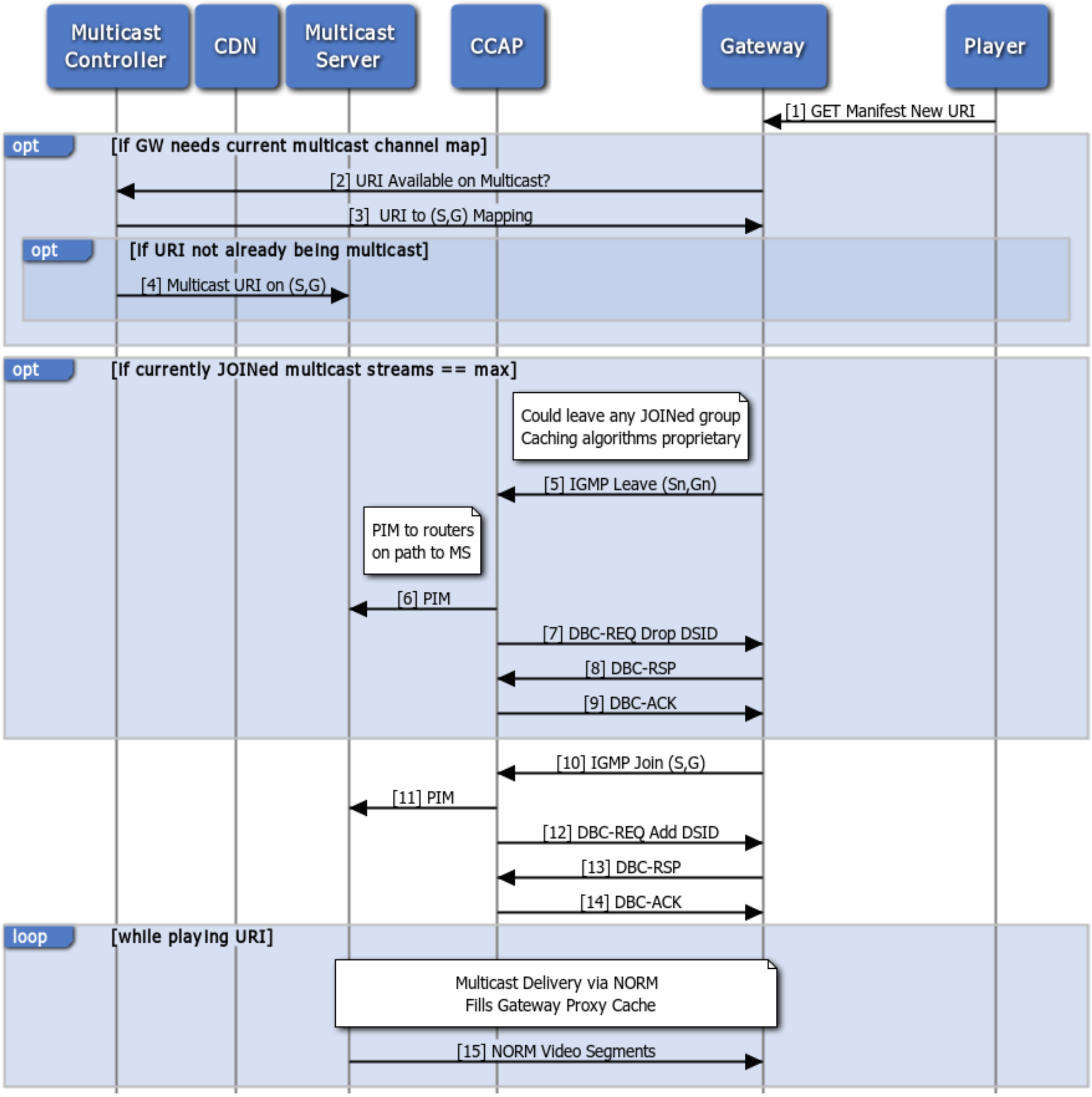

Figure 5 – Multicast Cache Filling

The unicast content delivery sequence and the multicast cache filling sequence are initiated by the same trigger - a Player request for new content. In the multicast cache filling sequence, if the Gateway doesn't have *a priori* knowledge of (a) whether or not content is available via multicast and (b) what the (S,G) for that content is, then the Gateway needs to determine by querying the Multicast Controller (as shown in steps 2 and 3). This request could potentially lead to the Multicast Controller deciding to offer this content on multicast and signal that to the Multicast Server (step 4).

An M-ABR-capable Gateway typically has a limited number of multicast receivers. There is one multicast receiver for each stream being received and each has an associated cache for that stream. A typical Gateway might have 5 multicast receivers and, thus, be capable of receiving 5 simultaneous multicast streams. Steps 5 through 9 have to do with freeing up a multicast receiver if they are all currently in use. Once a multicast receiver is available, Steps 10 through 14 show the Gateway joining the multicast group for the M-ABR stream (with both IGMP/PIM and DOCSIS messaging).

At this point the multicast receiver is available to start receiving video segments via NORM. These segments are cached and made available via the transparent proxy to fulfill requests from the Player. However, there is a bit of a race condition here - the Player is requesting segments sequentially (and potentially requesting segments faster than it is playing them) and the Gateway is also getting these segments delivered sequentially via multicast. How does the system increase the likelihood that there is a cache hit and ensure that segments have been delivered via multicast in advance of them being requested by the Player? This is where manifest manipulation comes into play. As mentioned previously, the Gateway typically knows about one more segment than the Player. The system uses this to provide the Gateway with a timing advantage over the Player. The goal is for the Gateway to have segment n waiting in cache when the Player requests it while, simultaneously, the multicast receiver for this stream is receiving segment n+1 and, thus, staying ahead of the Player's requests.

7.3.2 M-ABR Infrastructure Interactions

The following sequence diagram is intended to illustrate a general data flow (not individual message exchanges) which could be utilized to implement an M-ABR service.

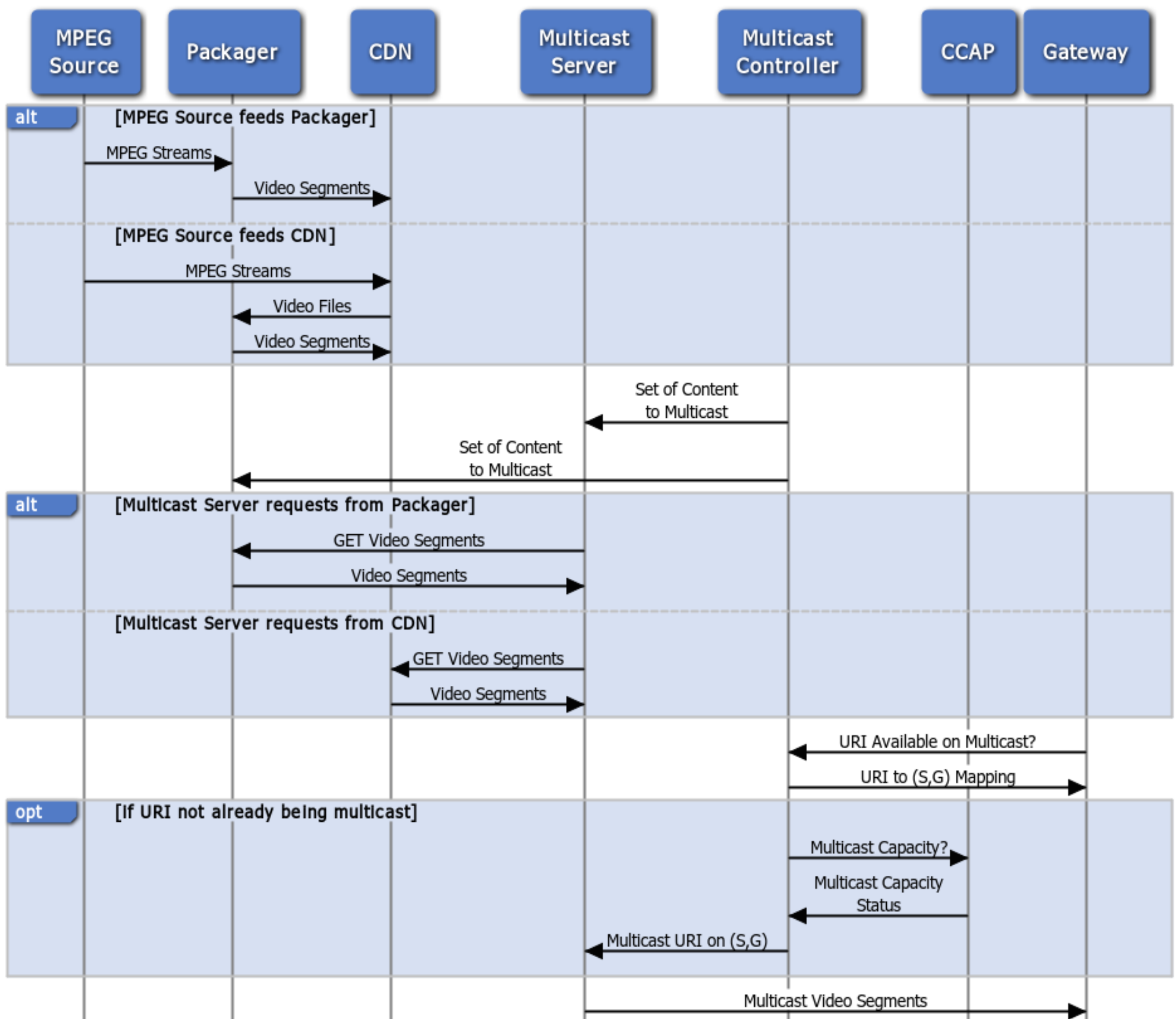

Figure 6 – Example High-Level Infrastructure Interactions

There are a number of ways a Multicast ABR infrastructure can be architected and, even with the same set of components, there are a number of different ways content can flow through the system. This sequence diagram shows a few of the most common alternative architectures. Video starts at the MPEG source and typically travels from there to either the Packager or the CDN. If it goes to the CDN first, then it needs to go to the Packager so that it can be segmented. In either case, segmented video should be available on both the CDN and the Packager (although, it may only be available temporarily on the Packager).

The next couple of steps in the sequence diagram illustrate the fact that the Multicast Controller dictates what content and bit rates are delivered via multicast. This is communicated to the Multicast Server and, potentially, to the Packager as well. Once the Multicast Server knows the content to multicast it can get that content. Again, some architectures have the MS get the content from the Packager and others from the CDN.

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 Section 8.2.2). 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 Section 8.2.1). The mechanism used to determine the content to multicast impacts the order of events in the sequence diagram (although, it is largely abstracted from the diagram).

The sequence diagram begins involving the Gateway in the last few exchanges. It shows the Gateway enquiring about the (S,G) for a given M-ABR stream and the Multicast Controller providing that data. This can then potentially trigger the Multicast Controller to communicate with the CCAP about the availability of multicast capacity (although, in a real implementation, this is likely an on-going item of synchronization). It can also trigger the MC to direct the MS to begin multicasting a given content stream which it may have content for, but is not currently multicasting to the requesting Gateway's serving group.

While the exact order of many of these steps is implementation dependent, the intent of this section was to provide a high-level overview of how these components might interoperate to provide an M-ABR service compatible with the player-oriented flows shown in Section 7.3.1.

8 MULTICAST CAPACITY-RELATED CONSIDERATIONS

The primary motivation for deploying IPTV over IP Multicast is improved efficiency over unicast. While time-shifting, VOD and other viewer behavior changes have created more and more unicast viewership, it is still the case that over 75% of tuners are tuned to live linear television in prime time [Eshet-14]; thus, the potential gains for multicasting are substantial.

8.1 Background & Overview

Why multicast? Many papers have covered this topic, but the fundamental reason to multicast is to take advantage of viewer behavior to maximize efficiency. There are different ways to look at it. [IMC-08] observes that "the top 10% of channels account for 80% of viewers"; [Horrobin-13] observed that during prime time the top 10 channels were viewed by 58% to 60% of viewers and the top 50 channels were viewed by 97% of the viewers. The bottom line is that while DVRs and time-shifting have had an impact on user behavior, a large percentage of users are still watching live linear TV and a small number of channels still command large viewership.

For multicast to be successful, the same packets for the same content have to reach multiple receivers at the same time. This is true of live linear TV, but not true of cloud DVR, VOD or other types of viewership where the viewer is either viewing different content than is currently being transmitted from broadcasters.

Best Practice: Multicast live linear TV.

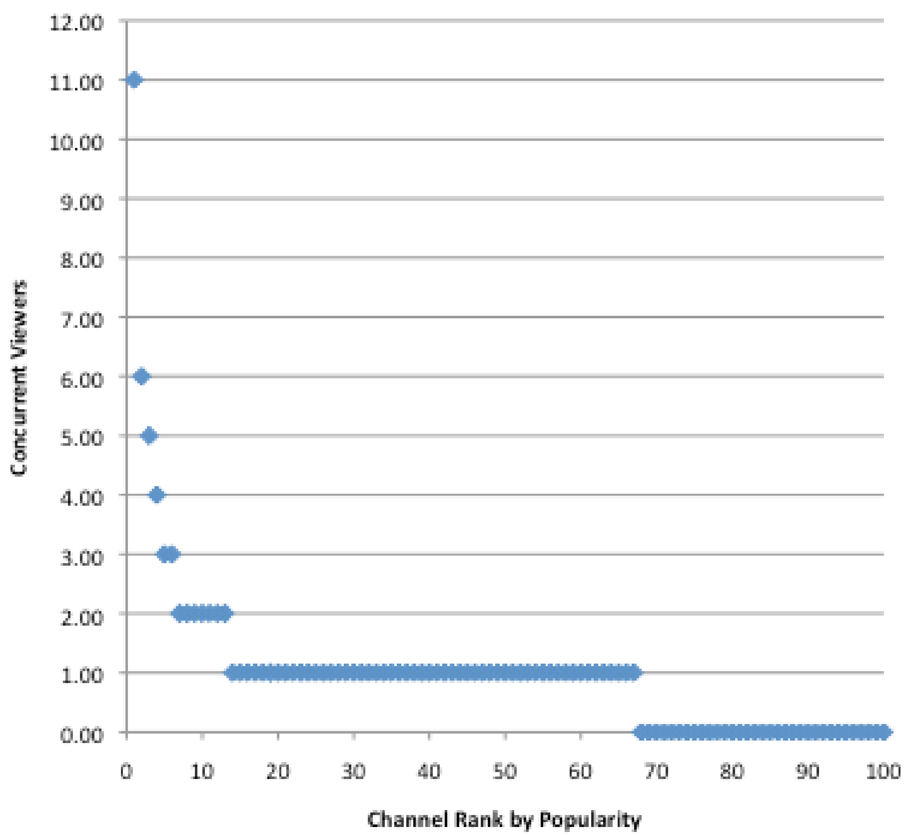

Figure 7 – Concurrent Viewers versus Channel Rank

Maximizing efficiency was the same motivation for the development of SDV. People often think of QAM-based SDV and IP-based multicast together, but they are really two different sides of the same coin. SDV maximizes efficiency in the "long tail" by not wasting broadcast QAMs on channels with no active viewers. Multicast, on the other hand, maximizes efficiency in the "tall head" by not wasting IP unicast capacity sending the same content to multiple viewers at the same time. Figure 7 uses a Zipf distribution with a shape value of 0.91, which is based on internal CableLabs data collection during 3 prime-time periods. This does not align completely with the Zipf-Mandelbrot modeling of [Ulm-09] and subsequent work, but this diagram is really just intended to illustrate the overall shape of this type of viewership data with its "long tail" and "tall head" - which exists in both models.

While the emphasis when discussing IP Multicast is generally on improved efficiency in the access network, it should be noted that multicast also saves capacity on metro/core networks that feed the CCAP. Just like in the access network, when unicast content retrieval is used, the CDN has to ship multiple copies of the same content to the CCAP. Thus, IP Multicast is important for reducing the capacity required to support IPTV in the aggregation network as well.

This section deals with the "what", "where" and "when" aspects of M-ABR. How does an operator deploying M-ABR best determine what content to provide via multicast and when to provide it via multicast? Once that content is identified, where should the operator multicast it? (These last of these questions is clear on the level of an individual node - content should be multicast only where it is requested, but this technical report deals with the "where" question in the context of sizing Serving Groups.)

8.2 Multicast Content Selection Approaches

The IP Multicast Working Group has identified two main approaches to determining what content to provide via multicast.

- Viewership Driven Multicast: any content with more than one simultaneous requester is multicast
- Policy Driven Multicast: *n* configured channels are available for request via multicast (typically, these are the *n* most popular channels for a given time period and location)

This section describes these two different approaches in more detail.

8.2.1 Viewership-Driven Multicast

Viewership-Driven Multicast is a term defined in this document that is intended to describe an approach to multicasting whereby the multicast controller determines what content is to be multicast primarily on the number of simultaneous requests for that content. This is sometimes referred to as the "greater than one viewer" model, which refers to the fact that (generally) in this model any content which has more than one viewer is multicast. Thus, this approach takes any opportunity to improve efficiency by multicasting.

These systems are very dynamic as they match multicast content to real-time viewership very precisely. These systems maximize multicast efficiency and preserve network capacity. This precision can come at the expense of complexity as these systems need to be prepared to multicast any content available to viewers. Another negative of this approach is that it is not "data friendly," meaning that it makes data traffic engineering more difficult on bonding groups where both video and data co-exist. The reason data traffic engineering may be more difficult in these cases is that there is no defined maximum amount of video traffic and, thus, video traffic could theoretically "starve" data traffic, or at least consume more capacity than predicted at the expense of data capacity - potentially creating congestion in the data portion of the bonding group.

8.2.2 Policy-Driven Multicast

Policy-Driven Multicast is a term defined in this document that is intended to describe an approach to multicasting whereby the operator determines a specific set of channels that are made available for multicast delivery. Typically, operators use viewership history during different time periods to determine the set of channels available for multicast at any given time. This is sometimes referred to as the "top n channels" model, which refers to the fact that in this model a maximum number of channels are made available for multicast. If a viewer requests a channel which is available on multicast, they are joined to the corresponding multicast feed; otherwise, they are given unicast video. Even if more than one viewer requests the same content, if it is not available in the current set of multicast channels, they receive the content via unicast.

This model is not entirely static as the set of channels available on multicast can change, but the changes to this set are driven by operator policy rather than real-time viewership. This allows the operator more control of both what content is available via multicast (both which channels and at what bit rates) and how much multicast is happening on their systems at any given time. Since the set of channels available for multicast is generally relatively small and changes relatively slowly, these systems are generally viewed as simpler. This simplicity potentially comes at the expense of efficiency, as some channels with more than one viewer can be unicast in this model. However, it should be noted, that if the number of channels available for multicast is sufficiently large and the operator's ability to predict channel popularity is good, then this approach can be very nearly as efficient as Viewership-Driven Multicast.

It should also be noted that just because *n* channels are available to be multicast doesn't mean that *n* channels are being multicast. A given channel is only multicast in a given serving group when at least one gateway in that serving group has requested access to that multicast channel by joining the corresponding multicast group.

8.2.3 Viewership & Policy Driven Multicast Hybrids

There are hybrids between these two models. One such hybrid is referred to in this technical report as "Viewership Driven with Maximum Number of Multicast Channels". As in Viewership-Driven Multicast, the set of channels multicast at any given time is driven by real-time requests for content. However, like Policy-Driven Multicast, there are a maximum number of channels that are allowed to be multicast.

This approach requires a system which can support the dynamism of Viewership-Driven Multicast, but, theoretically, this approach may not be as optimal as pure Viewership-Driven Multicast as there may be times where the number of multicast channels required to match the requested viewership exceeds the configured maximum. However, in practice, video traffic engineering can be utilized to ensure that this "multicast blocking" happens with extremely low probability (e.g., 99.9% non-blocking). Further, this simplifies data traffic engineering as the portion of the bonding group dedicated to data traffic has a guaranteed minimum value for its maximum capacity[5] rather than a statistical maximum capacity.

Another hybrid between these two models is "Viewership-Driven with Limited Bit Rates". In a pure Viewership Driven model, any stream with more than one consumer will be multicast regardless of bit rate. This hybrid model adds a policy component that limits the number of bit rates which are available for multicast. Typically in this model, bit rates are limited to HD-only or HD- and SD-only. This simplifies the Viewership-Driven model by reducing the amount of content that the Multicast Server has to be prepared to multicast. This improvement comes at the potential cost of reduced efficiency, but lower bit rates use less bandwidth and there are fewer consumers of these streams, thus, this hybrid is generally viewed favorably by operators in the Working Group.

8.2.4 Multicast Content Selection Approach Considerations

While Viewership-Driven and Policy-Driven Multicast have very different mechanisms, it is important to note that with some traffic engineering the efficiency differences between these two approaches can be minimized.

Figure 8 shows the number of individual viewers by channel rank. The red circle shows the viewership of the Top 10 channels. The blue circle shows the viewership of the Top 20 channels. The set of channels where there are at least 2 viewers, which would be the set of channels which would be multicast if using the Viewership Driven multicast model, contains fewer than 20 channels. Thus, in this example, a Policy-Driven Multicast approach where "up to" the top 20 channels were multicast and a Viewership-Driven Multicast approach would have identical efficiencies as they would be multicasting the same set of channels. By contrast, a Policy-Driven Multicast approach where "up to" the top 10 channels were multicast would somewhat be less efficient than the Viewership-Driven model where 13 channels have at least 2 viewers. However, both approaches capture the channels that provide the biggest gains in efficiency and, in this example, the efficiency improvement of the Viewership-Driven approach over the Policy-Driven approach when n = 10 is only 6.5%.[6]

---

[5] In this case the data capacity isn't fixed but a minimum value for the maximum data capacity is known. This is likely best illustrated by example. If an operator had a bonding group with 36Mbps of aggregate capacity and allocated 16Mbps to video that means that data would have a minimum maximum capacity of 20Mbps, which is to say, if the video portion of the bonding group was under-utilized the data capacity could exceed 20Mbps, but it would never be less than 20Mbps (which is the data capacity limit when the video capacity is fully utilized).

[6] With the Policy Driven approach and the top 10 channels, in this example, the multicast gain is 65.2% (10 multicast and 6 unicast streams instead of the 46 required for pure unicast). Whereas, the Viewership Driven approach would have a multicast gain of 71.7% (13 multicast streams instead of 46 unicast streams) so there is a 6.5% difference in multicast gain between the two different models in this example.

This section is intended to illustrate the point that these two methodologies, which differ substantially in complexity, can be similar in their effectiveness under the right circumstances. That said, the figure is based on theoretical behavior at a single point in time. Operators should look at the set of their Top N channels during different viewing periods to see how dynamic this set is. If the set is relatively predictable and stable, then a Policy-Driven channel selection approach could achieve acceptable levels of efficiency improvement without the complexity inherent in the Viewership-Driven model. It should also be pointed out that an operator could migrate from the Policy-Driven model to the Viewership-Driven model by only making modifications to the Multicast Controller, so there is potential for a relatively straightforward evolution from one approach to the other.

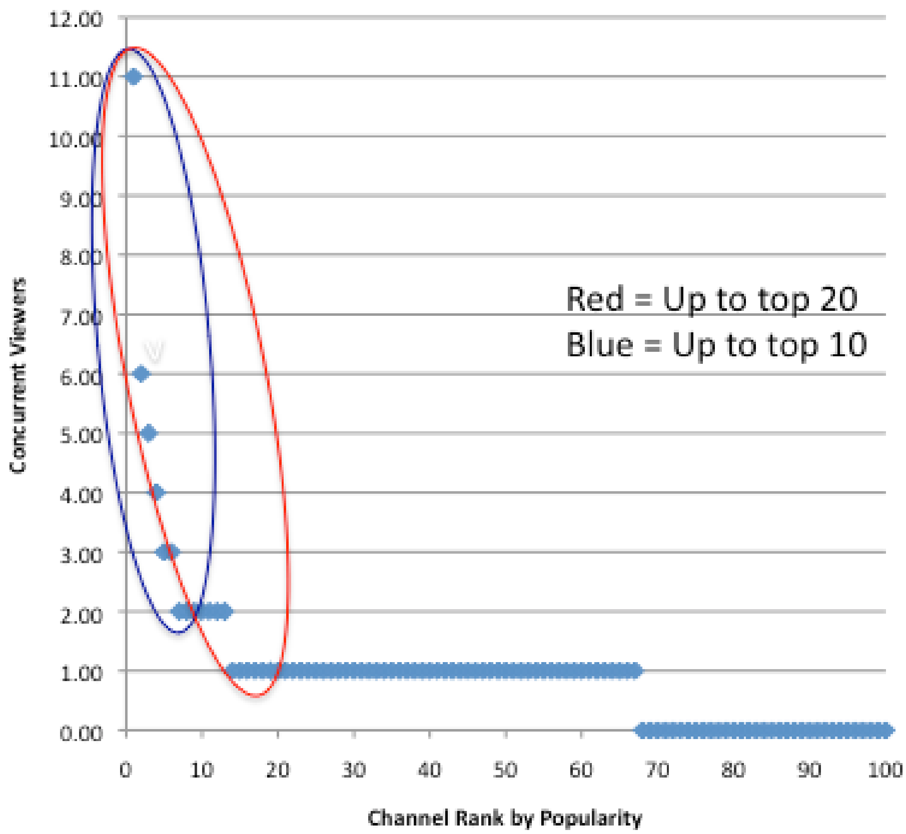

Figure 8 – Concurrent Viewers versus Channel Rank with Viewers of Top N Channels[7]

Whether Policy-Driven Multicast or Viewership-Driven Multicast is more efficient is entirely a function of:

- The maximum number of unique channels that a Policy-Driven Multicast system multicasts[8]
- How well the specific channels that a Policy-Driven Multicast system chooses to multicast match the actual viewer channel demand

Thus, choosing a multicast channel selection methodology should be based on other design considerations in addition to efficiency, such as system complexity, ease of traffic engineering, etc.

[7] This is the same figure as in Figure 7. Again using "best fit" numbers from 3 field measurement periods and a Zipf model. In this case the model is using 100 active viewers and 588 available channels.

[8] [Horrobin-13] indicates that 93-94% of live linear TV tuning during prime time is of the top 40 channels and 97% of tuning (at any time) is of the top 50 channels.

8.3 Serving Group Sizing Considerations

Another major design consideration in deploying M-ABR is Serving Group Size. In general, the larger the Serving Group then the greater the "multicast gain" [Ulm-09] [Horrobin-13] for video. However, if operators are utilizing multi-service bonding groups where both the data and video service share the same bonding group (refer to Section 9.1.1), there is a tradeoff which needs to be considered.

Data service is largely a unicast service, thus as Data Serving Groups get larger and serve more CMs/Gateways, there is more contention for a fixed capacity in that Serving Group. Thus, operators deploying multi-service Serving Groups need to balance the desire to maximize multicast efficiency/gain for their video service with the need to manage contention/utilization for their data service.

Typically a Data Serving Group is a single fiber node. Data Serving Group sizes have been coming down as usage has been increasing and as operators have looked to increase capacity. If operators maintain their Data Serving Group size and add IP Multicast Video Service to this same Serving Group, it limits the potential number of receivers of M-ABR streams, which reduces the potential multicast gain.

If operators use dedicated DOCSIS QAMs (or CCAP QAM replication) for their M-ABR service, then the Video Serving Group size can be increased by spanning multiple nodes (without negatively impacting the Data Service). However, this comes at the expense of no longer being able have data traffic consume unused video capacity and somewhat defeats the purpose of transitioning to "All IP" as the access network is not a true multi-service network.

8.4 Centralized versus Decentralized Multicast

Another question in the vein of "where" to multicast is whether the Multicast Server should be centralized or should be decentralized. In Decentralized Multicast, the Multicast Server is co-located with the CDN Edge Cache such that unicast ABR and M-ABR generally originate in the same facility. In Centralized Multicast, the Multicast Server is located in a more centralized location than the Edge Cache.

From a capacity perspective, there is very little difference between these two models. In both cases the capacity needed is proportional to the number of streams rather than proportional to the number of viewers as the CDN (which feeds unicast) gets one copy per stream just as the Multicast Server both consumes and delivers one copy per stream.

However, there may be advantages to centralized multicast - especially early in deployment where the number of potential M-ABR viewers is low. In this case, centralized Multicast Servers would reduce the initial hardware investment required to deploy this service. There are few other benefits to centralized multicast, perhaps simplified administration or deployment, but there are potential performance impacts from increased latency and the increased complexity of maintaining a larger multicast footprint. In general, in the long term, operators in the Working Group were looking towards decentralizing their Multicast Server deployments.

8.5 IP Multicast Migration

Another design consideration operators wrestle with is "when" to begin supporting M-ABR. To look at a degenerate case, if there is only one M-ABR capable Gateway on a node, then there is no advantage to multicast over unicast. If there are only two M-ABR capable Gateways on a node, if they both happen to be in use and tuned to the same channel at the same time, then there is an advantage to IP multicast. If one extends this logic, it becomes clear that given variability in viewership (with respect to time, channel and bit rate) there needs to be an established population of M-ABR capable Gateways to achieve any real efficiency benefit from M-ABR. While average M-ABR usage can be an improvement over unicast ABR at relatively low viewership levels (~10 active viewers), the set of channels viewed at any one time is extremely variable, thus for M-ABR to improve efficiency over unicast 99.9% of the time requires roughly 80 active viewers [Ulm-09].

The following section shows an approach to estimating the penetration level required for M-ABR capable Gateways and ABR players to achieve a given level of ABR Players tuned to live linear TV during times of peak usage. Both scenarios take a typical node size in households passed (HHP) as well as cable and digital cable take rates to calculate a number of Digital Subscribers per Node. To this number a "M-ABR Penetration" rate and an ABR Players per Sub" rate are applied. Clearly, these numbers are going to vary by operator and over time. Operators who migrate to IP Multicast video and aggressively deploy M-ABR capable Gateways will be able to achieve efficiency benefits from IP Multicast a greater portion of the time than operators who deploy more conservatively.

Table 2 – Multicast ABR Deployment Scenarios

Parameter	**Value**
Households Passed per Node	500
Cable Take Rate	60%
Digital Take Rate	85%
Digital Subscribers per Node	255
M-ABR Gateway Penetration	25%
ABR Player per Sub	1
ABR Players per Node	64
Peak Simultaneous Usage	70%
Linear TV Usage	80%
ABR Players Tuned to Linear at Peak	36

Parameter	**Value**
Households Passed per Node	500
Cable Take Rate	60%
Digital Take Rate	85%
Digital Subscribers per Node	255
M-ABR Gateway Penetration	75%
ABR Player per Sub	2.8
ABR Players per Node	536
Peak Simultaneous Usage	70%
Linear TV Usage	80%
ABR Players Tuned to Linear at Peak	300

Per [Ulm-09], the scenario on the left with 25% M-ABR Gateway penetration and only a single ABR Player per Sub (either IP-STB or companion device) would only save around 25% on average and would have no capacity savings due to multicast during times of high variability. The scenario on the right with 75% M-ABR Gateway penetration and multiple ABR Players per sub would have an estimated savings of over 60% on average and still around 50% during times of high variability. Thus, the level of penetration of both multicast-capable gateways and ABR Players tremendously impacts the potential multicast gain that operators can achieve.

Operators may want to consider the possibility of deploying M-ABR capable Gateways before even having an M-ABR infrastructure in place. Then, once penetration levels of M-ABR capable Gateways in the field is on track to provide multicast gain a significant portion of the time, an operator could begin deploying M-ABR infrastructure and turning up IP multicast video service.

9 MULTICAST DELIVERY DESIGN CONSIDERATIONS

In M-ABR, there are fundamental tradeoffs between IP multicast efficiency and latency, bandwidth efficiency, complexity, etc. For example, the recommended multicast transfer protocol (NORM) has the ability to utilize content-level FEC to minimize the potential for loss. Using NORM's FEC capability will add overhead to the content, but it may reduce the potential for multicast/unicast retransmission and increase overall efficiency.

Similarly, NORM has the ability to retransmit over multicast in case multiple receivers missed the same content transmission; however, using this capability can increase latency. Operators who want maximal multicast efficiency may want to retransmit over multicast at the cost of latency, but other operators may be willing to accept reduced multicast efficiency to minimize latency. Further, the potential for lost transmissions is a function of plant conditions, physical-layer optimization, transport-layer FEC and other factors. Thus, what is optimal may not only vary between operators, but might even vary across regions within the same operator.

This section explores the design considerations and tradeoffs related to the actual mechanics of M-ABR delivery. It is focused on the question: "How should operators best design their M-ABR systems to meet the service design goals detailed in Section 5.4?" Or, when there is no clear-cut best practice, "What design tradeoffs should operators focus on when designing their M-ABR systems?"

9.1 Access Network Design Considerations

The Best Practices in Access Network Design hinge on a single fundamental question:

Are video and data service going to share the same QAM channels?

This question is a combination of technical and philosophical, but does involve a fairly large tradeoff. Operators need to decide whether it is more efficient to increase multicast gain by having larger M-ABR-only Serving Groups or more efficient to integrate the video and data services such that capacity unused by M-ABR can be used by the data service.

Another design consideration for examining this tradeoff has to do with flexibility and service agility - operators need to decide if the service agility benefits of a truly multi-service network are worth the cost of potentially lower multicast gains which are inherent in an integrated multi-service network.[9] Finally, despite the disadvantage in service agility, there may be valid legal, policy or business reasons to maintain video on separate QAMs from other services.

9.1.1 Access Network Quality of Service

With Best Effort packet forwarding, it is possible that downstream bonding groups will occasionally experience congestion due to higher than average utilization. When bonding groups carry a mix of traffic for various services (as discussed further in the next section), traffic loads are even more unpredictable and short-term congestion events are expected. To some degree the operator can use traffic engineering practices to minimize the occurrence of these events, but it is not feasible to prevent them entirely.

The goal of M-ABR delivery is to significantly reduce the downstream traffic load for popular streams. However, if M-ABR segments arrive late or experience loss, this leads to retransmission of the missing or late data. If the content is extremely popular, the amount of retransmission traffic can be many times the amount of the original transmission. If such popular M-ABR segments are delayed significantly due to congestion of the bonding group, this could result in a cascade breakdown as all tuned Gateways experience cache misses, and revert to unicast retrieval, further exacerbating the bonding group congestion, and affecting all services.

Thus, for stability of the network and protection of all of the services that traverse it, it is extremely important that M-ABR traffic be delivered reliably and in a timely fashion.

[9] This is because data serving groups generally need to be smaller to keep utilization within operational bounds and to minimize contention. That said, Table 2 showed that the number of ABR Players Tuned at Peak can reach substantial numbers even with a node size of 500 HHP, but this is very dependent on the number of players per subscriber and the M-ABR gateway penetration. So, if future viewer behavior is consistent with that of today and penetration continues to increase, it is possible that, over time, even with node sizing optimized for data service, meaningful multicast gain can be achieved.

DOCSIS provides QoS on a per Service Flow basis. Thus, the following best practices have been identified.

Best Practice: M-ABR video should traverse dedicated multicast Service Flows.

Best Practice: M-ABR multicast Service Flows should be configured to provide enhanced Quality of Service over Best Effort traffic.[10]

DOCSIS provides two different mechanisms that can be used to ensure enhanced Quality of Service to the packets of a downstream Service Flow:

1. Traffic Priority - When a Service Flow is given elevated priority relative to other traffic, the CCAP scheduler will expedite delivery of the Service Flow's traffic.[11] As long as the total amount of traffic with the same or greater priority does not exceed the capacity of the channel set, delivery can be assured. So, along with traffic priority configuration, some traffic engineering and admission control would be needed for this mechanism to be successful.
2. Minimum Reserved Traffic Rate - When a Service Flow is configured with a Minimum Reserved Traffic Rate, the CCAP scheduler will provide highest priority treatment to the Service Flow as long as the actual traffic rate does not exceed the configured value.

QoS for non-multicast video traffic can have an impact on Channel Change Time and other QoE metrics. In particular, channel changes often involve unicast retrieval of the initial segments for the new channel. If these initial segments do not receive QoS guarantees, this could increase the channel change time. If the IP video service is intended as a legacy video replacement service, then QoS for unicast ABR traffic should be considered. However, QoS for unicast ABR traffic is beyond the scope of this technical report.

9.1.2 Multi-Service vs Dedicated Channels/Bonding Groups

At the beginning of this section, two fundamental questions were posed. The second of these was whether or not data and M-ABR video would traverse the same RF channels. This section addresses best practices and design considerations which come out as a consequence of that decision.

If an operator decides that they want to pursue a multi-service integrated network, then another question arises from this decision:

> *How do operators meet the design goal of reliably delivering M-ABR video when they are mixing M-ABR video into channels that also deliver data and other IP-based services?*

To answer this subsequent question, the next subsection looks at Service Coexistence.

9.1.2.1 Service Coexistence

When multiple services share the same channels/bonding groups, care needs to be taken that these services don't negatively impact one another. With voice and data service coexistence, this is done via admission control. After *n* calls are admitted, subsequent calls are rejected. This ensures a guaranteed minimum amount of overall capacity for the data service sharing the same bonding group as the higher-priority voice service. With content selection mechanisms that limit M-ABR up to a fixed number of channels, there is also an inherently guaranteed minimum amount of overall capacity remaining for the data service.

However, with pure Viewership-Driven approaches to multicast content selection, any number of channels can potentially be multicast. Since M-ABR requires higher-priority service than unicast ABR or Best Effort data, it is possible that M-ABR traffic can negatively impact the capacity available for the data service. Thus, the data service needs protections to ensure that it has a guaranteed minimum amount of capacity.

To address this issue, the IP Multicast OSSI focus team recently developed a mechanism called Admitted Multicast Aggregate Bandwidth, which is designed for ensuring managed coexistence of multicast and unicast traffic. Rather than creating a hard-limit for overall multicast within a bonding group (or MAC domain interface) on the CCAP,

[10] This does not necessarily mean that providing guaranteed QoS is also a Best Practice for unicast ABR. Some operators may want to provide QoS for unicast ABR while other operators may recognize that only a single viewer is impacted by the quality of this stream and want to rely on traffic engineering and the inherent adaptability of ABR to ensure QoE for unicast media streams.
[11] The specific treatment of Traffic Priority by the CCAP scheduler (e.g. strict priority, weighted fair queuing, weighted round robin, etc.) varies by CCAP implementation, so the operator should work with their CCAP vendors to understand the optimal approach for their specific situation.

this mechanism creates a new alert from the CCAP, which can be utilized to alert that a given capacity allocation threshold has been exceeded.

This alert can be used by the Multicast Controller to limit additional multicast in this bonding group. This approach adds complexity to the Multicast Controller, as it needs to know the association between multicast flows and channels/bonding groups. However, it also reduces traffic-engineering complexity (refer to Section 8.2.3) and ensures a clean segregation of data and multicast video services.

Design Consideration: If M-ABR and data services share the same channels/bonding group, is a statistical bound on maximum video usage sufficient or not? There aren't really pros and cons to this decision - either method can work. The question is which method is an operator's capacity planning group more comfortable with: a soft statistical bound on data and video capacity or a more deterministic "guaranteed" maximum capacity allocation for the video service (which simultaneously provides a guaranteed minimum amount of capacity for data service).

Design Consideration: As noted earlier, if M-ABR and data services share the same channels/bonding group, that means they have the same Serving Group size. This is challenging because for multicast efficiency, the larger the Serving Group the greater the efficiency. However, as the data service is a unicast service, the larger the Serving Group, the greater the contention for bandwidth. Thus, the needs of these two services are fundamentally at odds and operators will need to balance the desire for M-ABR efficiency with the need to provide sufficient capacity for the data service.

The Admitted Multicast Aggregate Bandwidth is the only standard DOCSIS mechanism for segregating services within a bonding group. That said, there may be additional vendor-specific mechanisms which can be used to achieve similar goals. However, such mechanisms are beyond the scope of this technical report.

9.1.2.2 Dedicated M-ABR Channels

If operators choose to use dedicated M-ABR channels/bonding groups to achieve greater multicast gain than is possible with smaller serving groups (especially in early deployments) then the issue of service co-existence goes away. The M-ABR video service and the data service co-exist by residing in separate dedicated channels/bonding groups. Thus, no admission control or other mechanisms are needed to ensure that these services do not negatively impact one another.

9.1.3 Dynamic Multicast Service Flow Creation

In Section 9.1.1 the technical report identified the Best Practice of using dedicated Service Flows for M-ABR traffic. What that section did not address was:

> *How should operators create DOCSIS Service Flows specifically for M-ABR traffic?*

There are two primary mechanisms that can be utilized. DOCSIS 3.0 provides a feature called Multicast Join Authorization that can dynamically create new Group Service Flows when triggered by an IGMP Join. PacketCable Multimedia also has the capability of creating multicast Service Flows dynamically.

Group Service Flows are a simple mechanism for creating QoS-enabled multicast Service Flows. There is a set of rules on the CCAP that indicate (a) which devices are authorized to join which multicast groups and (b) the QoS which should be used for different multicast groups. When an IGMP or MLD message arrives indicating that a given device would like to join a multicast group, the CCAP first performs an authorization check for the requesting device and multicast group. If this check succeeds, then the CCAP adds a DSID for this multicast group to the bonding group of the requesting CM/gateway and generates an upstream PIM request for the multicast content. When the multicast content begins to flow, its addressing is checked against provisioned classification rules for multicast traffic. The multicast traffic gets the QoS associated with the highest-priority Group Classifier Rule that matches the traffic.

Design Considerations: PacketCable Multimedia is a more flexible mechanism for creating QoS-enabled multicast Service Flows. Thus, it is likely a better long-term mechanism for establishing and controlling IP multicast Service Flows. However, many operators have business reasons for avoiding PacketCable Multimedia and their initial needs for IP multicast Service Flows can be met via the simpler Multicast Join Authorization/Group Service Flow mechanism.

Given that different operators have different design goals and business constraints there is no clear-cut best practice for dynamically creating multicast Service Flows.

9.1.4 Channels per Bonding Group

Unicast video can generally be streamed using a Gateway's existing Bonding Group (unless there are load balancing or other considerations). However, it is possible to deliver multicast video within the frequency range of a given Gateway's wideband tuner, but not as part of the Gateway's current bonding group. To access this content the Gateway needs to stop receiving a current channel and start receiving another.

Whether a given multicast session is being transmitted on a channel which is part of a Gateway's current bonding group or transmitted on a channel which is not part of that bonding group, a DBC exchange is required to join that multicast session. However, to add or remove a downstream channel from a bonding group takes additional time for retuning the Gateway's receiver and acquiring the new downstream channel. Thus, any DBC operation that requires retuning takes longer than one which does not.

As described in Sections 9.3.4 and 9.3.5, unless an M-ABR channel is cached, channel change time is largely a function of unicast behavior. Thus, using a DBC to gain access to a multicast stream will not increase the channel change time, but will increase the time to multicast.

Best Practice: To minimize time to multicast, all multicast channels for a given Serving Group should be in the same bonding group.[12]

This Best Practice may seem unrealistic - how could it be possible to have a bonding group which contains all of the M-ABR channels? Aren't a lot of channels going to be required? The reality is that looking at the maximum number of SDV channels watched and how many were unicast versus multicast, [Eshet-14] showed that only 5 to 10 channels are required to support even very large numbers of tuners.[13]

Table 3 – Worst Case Unicast vs. Multicast QAM Requirements per [Eshet-14]

# Tuners	Unicast (Max DS)	Multicast (Max DS)
125	10	5
250	17	8
500	35	10

These findings have broad implications. Not only does it give operators a sense of the maximum number of channels that will likely want to be multicast (i.e., back office sizing), but it also gives operators input on the number of demodulators they will need on their gateways to achieve this Best Practice. For example, a 24-downstream Gateway could have 8 QAMs tuned to channels that are primarily used for M-ABR video (but opportunistically contain data) and still have 16 QAMs available for data and unicast video service.

9.1.5 Access Network Security

This subsection is less concerned with service co-existence and service quality and more concerned with another Access Network function - security or, more precisely, privacy. DOCSIS provides a Baseline Privacy Interface (BPI) that is utilized to ensure that a hacked CM cannot be utilized to snoop a neighbor's traffic on the DOCSIS RF network. This is essential for unicast traffic, but given that ABR video uses end-to-end encryption, it is not necessary in this case.

[12] This is a little bit of an oversimplification. Multicasting multiple channel lineups could warrant a different approach. For example, an operator might have two video tiers – a basic tier and a higher tier that includes basic, plus additional channels. In this case, it might make sense to have one bonding group for the basic tier and a second bonding group for the higher tier, but the principle remains that to maximize QoE operators should avoid situations where CM's need to retune to access channels which are part of their channel line up. Note, this may mean that an operator might have, for example, a 24-channel CM for one tier and a 32-channel CM for another.

[13] The following table demonstrates the potential savings from using a multicast based Linear TV solution. It was derived from an analysis of Live viewership data collected 24x7 from 10,000's of STB over a month long period. It shows the maximum upper limit for various Service Group sizes with dozens of SG at each size. Note that there were ~2.75 STB per subscriber for this data. Actual bandwidth capacity may vary dependent on a number of key variables including total programs being offered, SD/HD/UHD viewership mix, SG sizes; but the relative savings from using multicast should be similar for other configurations.

Best Practice: Group Service Flows utilized for transporting M-ABR streams should disable baseline privacy and other access-layer security.

This approach reduces encapsulation overhead and the amount of processing required for encryption and decryption within the overall system.

9.2 Multicast Functional Design Considerations

This section addresses design considerations more directly related to the multicast function itself - aspects of multicast such as group membership and transport protocols are discussed.

9.2.1 Multicast Group Membership Design Considerations

IETF specifications for IPv4 multicast are long established and well understood. IPv6, while somewhat newer, utilizes multicast as a core part of its functionality and, thus, provides robust multicast capabilities. Some operators, however, are using IPv6 widely in the access network while still utilizing an IPv4-based multicast infrastructure. In this scenario, MLDv2 would be used in the access network, but this inherently IPv6 protocol would need to specify IPv4 multicast groups for membership.

Open Issue: How should gateways utilize MLDv2 (an IPv6-centric protocol) to joint IPv4 multicast groups?

Members of the IP Multicast Working Group are working with the IETF to address this issue.

9.2.2 Multicast Transport Layer Functional Design Considerations

Many reliable multicast protocols have been developed over the years, but few, if any, have been widely deployed. Certainly, none have been deployed on the scale that is being discussed by cable operators for M-ABR deployments. Initially, one operator developed a protocol. However, lack of standardization pushed them to look at other protocols with more standards-body support.

After careful consideration of many options, the industry has identified NACK-Oriented Reliable Multicast (NORM) [RFC 5740] as the protocol of choice for delivery of M-ABR. The primary reason for this is that NORM is extremely flexible. For example, it can be utilized to transfer data streams or files, with or without FEC and with just error detection or also multicast retransmission. Full discussion of the selection of NORM or a comparison to other reliable multicast protocols is outside the scope of this technical report.

Best Practice: Utilize the NACK Oriented Reliable Multicast (NORM) protocol as the multicast transport protocol.

9.2.2.1 FEC

The first operator decision regarding NORM is whether or not to use NORM's Forward Error Correction feature [RFC 5052]. The pro of using FEC is that it can reduce the amount of traffic required to replace errored packets - a single parity packet can "protect" multiple data packets. In the case where there are multiple errored packets within a single FEC block, it is possible that a single FEC packet can "fix" more than one.

Best Practice: Utilizing NORM FEC is a recommended practice for reducing repair traffic.[14]

The next decision facing an operator is whether to use proactive or reactive FEC. In the first case, FEC parity packets are always sent in addition to content packets - whether they are needed by any receiver or not. In the second case, FEC parity packets are only transmitted upon request. Clearly, proactive FEC has the cost of constant overhead. However, when properly tuned, this overhead can be fairly minimal (e.g. less than 5%) and, when combined with a fairly robust physical layer, can virtually eliminate the need for repair traffic. Whether unicast repair or multicast repair is used, proactive FEC leverages multicast, thus it can repair errors in different parts of the same transmission at multiple gateways.

Design Consideration: Operators should decide whether the overhead cost of multicasting proactive FEC is less than the cost of the corresponding repair traffic. Factors such as the amount of FEC needed to hit a target error rate under given conditions and whether unicast or multicast repair is in use could impact this decision. Luckily, NORM

[14] If an operator is using reactive FEC and using unicast repair, then using FEC may or may not be a best practice. In this situation the CDN would need to be able to send parity packets in addition to "pure" content packets, which is an additional requirement on the CDN solely to enable the potential reduction in repair traffic that FEC provides.

should make it fairly easy to perform measurements in networks of different quality to determine the answer to this question.

The final high-level decision regarding FEC for operators is - how much FEC protection should be used? The answer to this question likely not only varies across operators, but across regions at the same operator or across systems within a region. It is up to operators if they want a single operational parameter or if they want to vary this across their footprint.

People tend to talk about Codeword Error Rate (CER) as though it is a parameter of the channel and constant across the channel, but the reality in an HFC network is that CER varies by CM. This is because each CM has a different location and each individual drop can have different noise characteristics, which can impact the CER. To monitor the CER, operators need to use SNMP to poll their CMs for their Unerrored, Corrected and Uncorrectable Codewords.[15] These are continuous counters, so at least two samples need to be taken so that a delta can be calculated. The CER is the (delta Uncorrectable) / (delta Unerrored + delta Corrected + delta Uncorrectable) where the counters are read at the same time. A single uncorrectable codeword error will cause an entire packet to be dropped. Thus, codeword error rate and packet loss rate can be correlated.

NORM uses Reed-Solomon FEC as described in [RFC 5052] and [RFC 5510]. NORM sizes code blocks (k) and parity (n - k) in units of packets. The following table assumes a maximum payload size of a NORM sender message to be 1500B[16] and a video segment size of 2MB (which is 2 secs of an 8 Mbps stream).

Table 4 – NORM FEC Protection & Overhead Scenarios

FEC Block Size (k) (pkts)	200	200	200	252	400[17]	800
FEC Parity (n-k) (pkts)	20	10	2	2	2	2
% of Packets Protected w/o Re-TX	5.0%	2.5%	0.5%	0.40%	0.25%	0.125%
FEC Block Size (KB 1000B=1KB)	300K	300K	300K	378K	600K	1,200K
FEC Blocks per Video Segment (pkts)	7	7	7	6	4	2
Overhead %	10.5%	5.25%	1.05%	0.90%	0.60%	0.30%

CableLabs obtained data from 5 nodes in the network of a leading operator monitored over 24 hours and calculated the codeword error rate for each CM in the node every 50 minutes. These 24 per-CM samples were then averaged to get a CM-by-CM CER for the node and these per-CM hourly averages were also averaged to get an average CER for the node.

Table 5 – Codeword Error Rate

CMs in Node	Avg CER by CM
113	2.2014E-07
49	4.5036E-06
72	1.2905E-07
75	9.4050E-09
177	1.2778E-04
Average	2.6529E-05
Avg (drop best & worst)	1.6176E-06

The 177 CM node has by far the worst average with a CER of 0.0128%. This was due to four highly disadvantaged CMs that experienced uncorrectable codeword errors in every hour of the sample period. These worst CMs had an average hourly CER of (1.3%, 0.54%, 0.39% and 0.0004%), but they were very atypical, as 78.5% of the CMs in this same node did not have a single uncorrectable codeword during the 24-hour sample period. Assuming an 8

[15] These can be found in the docsIfSigQExtUnerroreds, docsIfSigQExtCorrecteds and docsIfSigQExtUncorrectables.
[16] This is likely not an ideal value as NORM with FEC has ~32B of overhead and, thus, such a segment would not fit in a single DOCSIS frame, but this number is useful for illustration.
[17] [NORM-DEV] states "The maximum blockSize allowed by the 8-bit Reed-Solomon codes in NORM is 255, with the further limitation that (blockSize + numParity) <= 255." This would preclude larger values examined in this table. However, in limited experimentation with these larger values, they appeared to work.

Mbps HD stream on a 256 QAM channel, this results in an approximate PLR of 16.0%, 6.7%, 4.8%[18] and 0.0049% for these 4 worst performing modems or an average PLR for the node of 0.15%.

Operators need to decide if they want to design their FEC for the worst-case CMs or average CMs or the worst case node or the average node. Operators should also decide how much retransmission traffic is acceptable and look at the tradeoff between retransmissions and FEC overhead. That said, Table 4 shows that in this admittedly small sample, the packet loss rate of the worst CM would not have been repaired even adding 10% FEC overhead. However, the loss rate of all but the worst 3 CMs in this node would have been repaired using 0.6% FEC overhead.

This analysis assumes that the occurrence of uncorrectable codewords is uniform. Given the bursty nature of noise in HFC plants, this is not a realistic assumption. For this reason, field testing needs to be done before more authoritative analysis can be performed and best practices can be identified. However, operators should be reassured in knowing that the amount of FEC overhead needed to protect the vast majority of M-ABR traffic is likely less than 1% and that even having 4 FEC parity packets per video segment file can provide significant protection.

9.2.2.2 Multicast Repair versus Unicast Repair

NORM is a very flexible protocol. It allows for modes where errors in a stream can be repaired as part of the multicast transport protocol itself, but it also allows for modes where errors in the stream are only detected by receivers and repaired independently of NORM. In this technical report, these two models are respectively referred to as Multicast Repair and Unicast Repair.

Multicast Repair is more efficient when multiple receivers have the same error, as only a single repair packet needs to be sent to provide this repair to all of the errored receivers. (Unicast repair, in contrast, would need to send separate repair packets to each errored receiver individually.) However, multicast repair is not deterministic - if guaranteed QoS parameters are set based on a given per stream data rate, then multicast repair traffic will create a flow which violates these boundaries. With multicast repair, one highly errored receiver can increase delay for the entire group of receivers acquiring a given stream (video retransmissions need to be timely to be useful and inserting these retransmissions into the stream can delay other packets in a CBR stream). Further, a malicious listener could potentially intentionally increase a stream's delay - pretending to be a highly errored receiver - unless security measures were put into place to prevent this.

Unicast Repair is as efficient as multicast repair when errors across receivers of the same stream are uncorrelated - i.e., it is uncommon that multiple receivers are unable to receive packets in the same FEC frame. Unicast repair may also be desirable in an environment where repair is uncommon as multicast repair has scalability and security concerns which unicast repair does not. The only disadvantage to unicast repair is the potential for inefficiency compared to multicast repair in environments where errors among receivers are highly correlated.

To get some insight as to whether the efficiency gains of Multicast Repair are worth the additional complexity it requires, an analysis was performed of downstream uncorrectable codeword error rates. This analysis was performed on a small set of pre-existing data. This data consists of CER measurements for a 24-hour period on 5 different nodes. For the first 10 minutes of every hour, codeword error measurements were made every minute. Then a single 50-minute measurement was made for the remainder of the hour.[19] The goal of the analysis was to determine the number of CMs with errors in the same time window. While this wouldn't guarantee that errors occurred for these CMs on exactly the same codeword, if there was no correlation even within the same minute, then the same codeword definitely was not errored for modems without an error.

Figure 9 shows on an absolute basis the number of CMs with errors in the same minute during the day. Figure 10 shows this same data on a percentage basis. It can be seen that on both an absolute basis and a percentage basis the 1-minute correlation of errors across CMs was low. When errors did occur, the Codeword Error Rate was relatively low (from the perspective of the number of packets which would need to be retransmitted); when more than 3 CMs had codeword errors in the same minute, the average codeword error rate amongst the errored CMs was 0.005%. Similarly, when the percent of CMs with codeword errors exceeded 6%, the codeword error rate amongst the errored CMs was 0.015%.

[18] It should be noted that these worst performing CMs are likely out of spec for their service and probably should be repaired.
[19] This is the same data set used in the previous section for the FEC analysis.

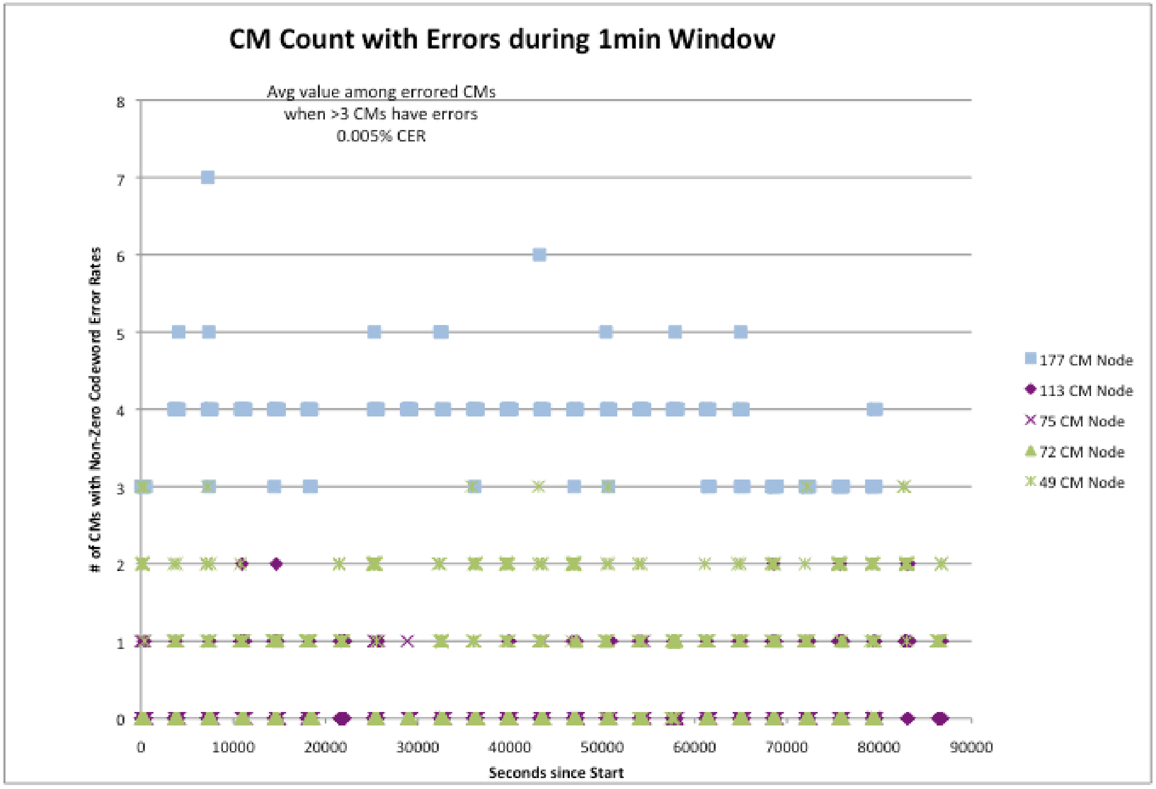

Figure 9 – CM Count with Downstream Codeword Errors (1 min.)

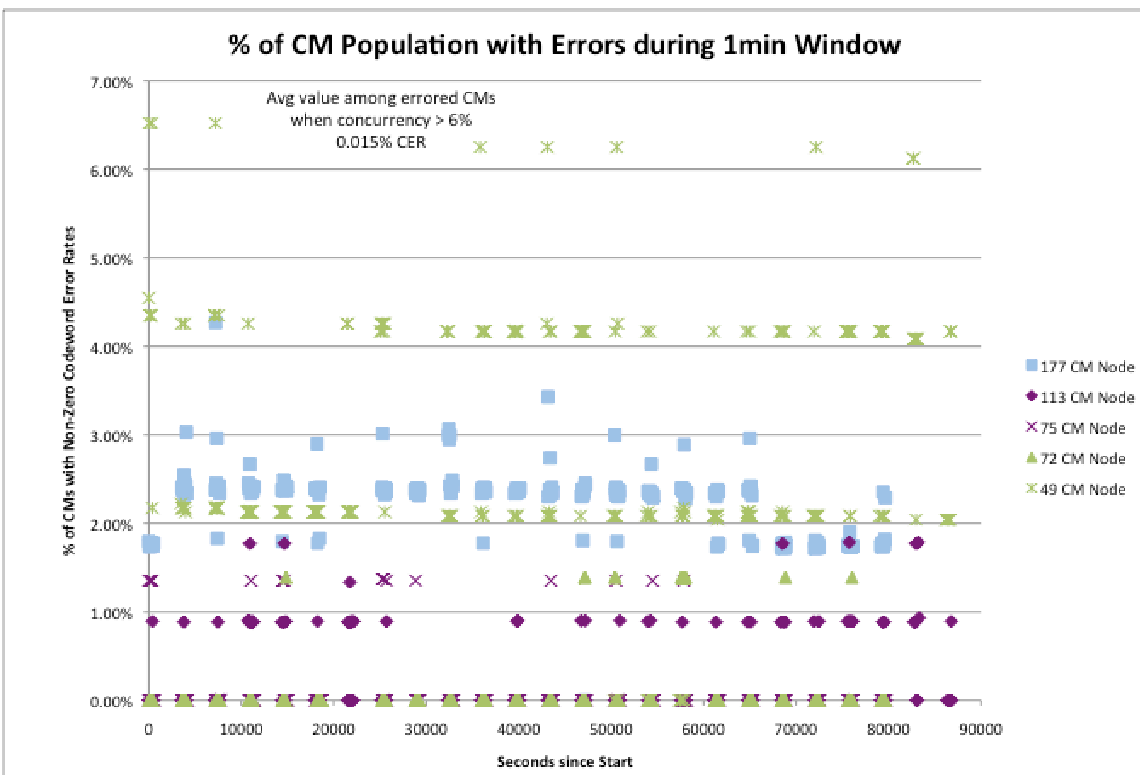

Figure 10 – CM Percentage with Downstream Codeword Errors (1 min.)

The focus of this analysis was on the 1-minute data as error correlation across 1 minute is more likely to indicate an error to the same code word than error correlation across 50 minutes. However, the 50-minute data was also examined. While it is unclear the degree of correlation on a shorter timescale, this data did show one outlier which likely warrants additional investigation or, at the very least, design consideration. On the sample date, one of the 5 sample nodes had one 50-minute interval where 69% of the CMs in the node experienced a codeword error. This is shown in Figure 11. While the average codeword error rate amongst CMs experiencing an error in this period was very low, 0.0028% this reveals the potential for infrequent bursts of higher codeword error rates. Servers facilitating unicast repair should be sized with these peaks in mind.

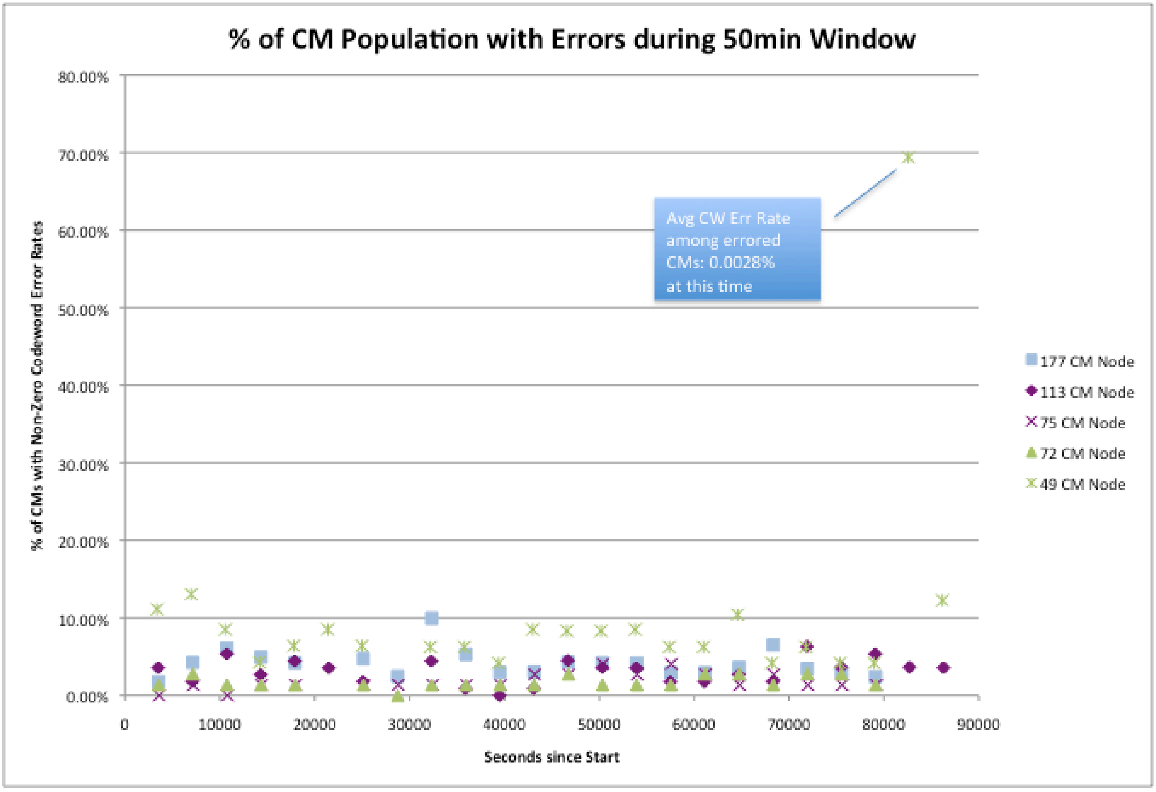

Figure 11 – CM Percentage with Downstream Codeword Errors (50 min.)

This extremely limited data set shows that the number of CMs with potentially correlated errors is relatively small compared to the overall population of CMs. Thus, the primary benefit of Multicast Repair - its ability to fix issues with multiple receivers simultaneously - may not apply in reasonably maintained DOCSIS networks. Further, there are negatives to using Multicast Repair that are clearly avoided when using Unicast Repair.

It should be noted that these measurements were all at the physical layer; transport layer FEC reduces the amount of unicast repair traffic required to correct for codeword errors at the physical layer. Pro-active FEC can substantially reduce the need for unicast repair traffic as transport-layer parity data received with the transmission can often be used to replace missing data without retransmission.

While operators should conduct their own measurements and analysis, the working group has identified unicast repair as a tentative Best Practice.

Tentative Best Practice: Utilize unicast repair when errors occur in data delivered via NORM.

The efficiency of unicast repair depends on its implementation. If an entire video segment file is downloaded when just a single NORM packet was undelivered, then unicast repair can be hugely inefficient. The drop of a single ~1500B packet could cause the retransmission of an entire 2 MB segment of video (2 second segment of 8 Mbps HD video). However, a more optimal implementation can recognize the missing portion of the data and utilizes

HTTP's Range-Request mechanism to fill in exactly the piece of content that is missing. (This requires both server-side and client-side support.)

Best Practice: The Gateway should utilize HTTP Range-Requests for repairing missing video segment data.

9.2.2.3 HTTP Header Acquisition & Delivery

As discussed previously, the Gateway acts as a transparent caching proxy for the Player. This requires that the Gateway support HTTP for retrieval by the Player. If the Gateway is proxying unicast content, then the Gateway can essentially pass the HTTP headers from the CDN through to the Player. However, if NORM is used simply to deliver video segment files, then this header information can be lost.

The first step in ensuring that HTTP headers are available for M-ABR content is to see that these headers get to the Multicast Server. A simple way to achieve that is to have the Multicast Server pull content from Origin/CDN or Packager via HTTP as though it were a Player. Thus, the Multicast Server has the appropriate HTTP header information available to deliver to the Gateway via multicast.

Best Practice: The Multicast Server should pull its content via HTTP so that it can deliver the appropriate HTTP headers to the Gateway.

Once at the Multicast Server, the challenge becomes how does the HTTP header information get delivered to the Gateway? As the MS is going to be delivering video segments via NORM, the most straightforward mechanism is to also utilize NORM to deliver these headers. Conveniently, NORM provides a message called the NORM INFO message that can be utilized to send metadata about files delivered via NORM DATA messages, and HTTP headers can be thought of as metadata.

Best Practice: The Multicast Server should utilize NORM INFO messaging to deliver HTTP header info associated with a given video segment such that the Gateway can reassemble the full HTTP response from the Origin/CDN for the Player.

9.2.3 Multicast Address Determination

In M-ABR there are at least two mechanisms relating to content addressing - the channel lineup shown to the subscriber via the Player GUI and the Multicast Channel Map (MCM) utilized by the Gateway to determine the (S,G) of multicast content. The channel lineup data includes the URI of a given video asset. The Multicast Channel Map provides the mapping between a video asset URI and the (S,G) of that asset, if that asset is being multicast.

Video assets can be accessed via unicast without knowledge of the (S,G) of the multicast stream for the asset. Thus, customer-facing functions such as channel changes and initial content "tuning," which rely on unicast access initially, do not have a dependency on access to the current Multicast Channel Map. Therefore, while delay in access to the MCM will negatively impact metrics important to operators such as time to multicast and multicast efficiency, timely access to the MCM does not impact customer quality of experience.

However, given that delays for the retrieval of the MCM will increase time to multicast and reduce multicast efficiency, it is still desirable to minimize the time it takes to acquire the MCM. Delivery of the MCM in advance of its being needed provides the best possible performance, as the acquisition time for the MCM is non-existent.

Best Practice: The Gateway should ensure that time for acquisition of the Multicast Channel Map does not increase time to multicast.

There are multiple ways that a Gateway could acquire the MCM without increasing time to multicast. One way to achieve this would be to deliver the MCM in advance of its being needed by delivering it periodically. This could be done either via multicast delivery or unicast retrieval. For example, a carousel-style approach could be used for periodic multicast delivery, while unicast retrieval could be incorporated as part of a "heartbeat" exchange between the Gateway and the Multicast Controller.

It should be noted that the first several segments of uncached media streams are always fetched via unicast. Given that video segments are on the order of seconds long and that several segments will likely be requested by the Player upon receipt of the manifest, the Gateway may have several seconds available to acquire the URI to (S,G) mapping without increasing the time to multicast. Therefore, a more transactional model whereby the multicast channel mapping of a single video asset is determined synchronously, on-demand may not negatively impact the time to multicast. Thus, there appear to be several diverse ways for the Gateway to determine the URI to (S,G) mapping for a video asset while still conforming to the Best Practice of not increasing the time to multicast.

Design Note: URI to (S,G) mappings can be delivered in advance (via unicast or multicast) or more transactionally, but, with care, any of these approaches can be used without increasing time to multicast.

Both unicast and multicast advance-MCM delivery mechanisms conform to the Best Practice by not increasing the time to multicast, but multicast MCM delivery consumes less capacity. With multicast delivery only a single copy of the MCM needs to be transferred to the M-ABR Serving Group. However, if a "heartbeat" exchange between the Gateway and the Multicast Controller is necessary for other reasons, then operators may consider the additional capacity used by delivering the MCM as part of this exchange as insignificant. This may depend on the number of channels an operator is multicasting – if the MCM contains the URI to (S,G) mapping for 40 channels then it will clearly be much smaller than if the MCM contains the same information for 400 channels.

Another design consideration around MCM delivery has to do with the rate of change in the channel map content. For example, an operator using Policy-Driven Multicast Content Selection might only change the channel map 2 or 3 times a day. Thus, whether the channel map is delivered via multicast or unicast is less significant as the difference in the amount of overhead is likely less substantial.

Design Consideration: The size of the MCM and the frequency with which it changes should be factored into the determination as to whether multicast or unicast delivery of the MCM is preferable.

9.3 Video Design Considerations

There are a couple of key metrics that underlie many of the video design considerations. These are:

Channel Change Time - The time from a viewer initiating a channel change to the new channel being displayed. (This is also sometimes referred to as the Zap Time.) This is a QoE metric for viewers, which, in some ways, has little to do with IP multicast as the initial ABR segments are often retrieved via unicast.

Time to Multicast - The time from a viewer initiating a channel change to the arrival of the first segment over multicast transport. This is a network efficiency metric, which has no QoE implications.

Time to Multicast has no QoE implications because the overall system is designed such that multicast content is cached on the gateway and any content requested by a player that is not in the cache is retrieved via unicast. Thus, a time to multicast of 10 seconds versus a time to multicast of 6 seconds only means that that there were 4 additional seconds where content that presumably could have been delivered by multicast, was retrieved via unicast instead - in both cases the Channel Change Time could have been 2 seconds (or less) and the viewer experience identical.

9.3.1 QoS & Video Delivery Rate

As mentioned in Section 9.1.1, one mechanism for providing guaranteed QoS in the downstream direction in DOCSIS is the Minimum Reserved Traffic Rate. This mechanism provides a Committed Information Rate for M-ABR streams. Section 9.1.1, however, did not discuss the potential for M-ABR streams to burst above their CIR. With unicast ABR, clients request video segments as quickly as they can be delivered - this rate often exceeds the actual bit rate of the content and leads to bursts above a minimum CIR required for the traffic. Burstiness in traffic can lead to congestion as peaks from several video streams can align and induce delay or loss.

Multicast delivery does not use TCP and, thus, does not have the inherent congestion control mechanisms that unicast ABR delivery does. NORM (and most multicast transport protocols) provides mechanisms for addressing this. In particular, the NORM allows for constant bit rate transmissions. The bit rate selected should not exceed any Maximum Sustained Rate assigned to the associate Group Service Flow or packets will be dropped/delayed.

Best Practice: The groomed rate of M-ABR streams from the Multicast Server should not consistently exceed the Maximum Sustained Traffic Rate, if any, on the associated Group Service Flow.

Best Practice: The Minimum Reserved Traffic Rate assigned to a Group Service Flow should be greater than or equal to the groomed rate of the associated M-ABR stream from the Multicast Server.

9.3.2 Manifest Manipulation

In standard ABR delivery, when first tuning to a channel the Player receives a manifest file and begins downloading the media segments listed in the file and playing them back. In multicast ABR delivery, the Gateway needs to be ahead of the Player such that an entire segment file is present in the Gateway cache and can be delivered to the Player upon request.

To do this, the M-ABR system typically performs "manifest manipulation" and removes the URL of the last segment from the manifest it retrieves from the CDN before sending the manifest to the Player. This avoids a race condition between the Embedded Multicast Cache and the player by allowing approximately one segment's worth of time for a segment to arrive via multicast and be readied for retrieval from the Gateway cache. Manifest manipulation can be performed either by the Gateway itself or in a back-office component. Some operators simply maintain two versions of the manifest – one for Players and one for other system components.

Best Practice: The system should have the capability of modifying or managing manifests to allow the Gateway's Embedded Multicast Cache to stay at least one segment ahead of a Player's requests.

9.3.3 Reception Caching & Predictive Tuning

Best Practice: Gateways should have multiple multicast receive buffers so that they can proactively receive and cache channels which the viewer might want to watch in the future.

This practice improves both the Channel Change Time and the Time to Multicast at the expense of minimal additional storage and complexity. There are many different algorithms for determining which channels should be cached. For example, the 4 most recently watched channels could be cached and, when one of these caches is needed for a new channel, the least recently watched channel could be ejected from the cache.

One can even envision predictive tuning of cached channels. This could be based on usage history, current popularity or user behavior (e.g., linear channel changing). There are many potential algorithms, and refinements to the algorithms to improve channel change times are likely to be a topic of research for years to come.

Best Practice: The Multicast Controller should know that a Gateway is "Tuned but Not Viewing" so that it can determine when to potentially terminate the multicast of a given stream.

This practice works with the previous practice to ensure that these performance improvements also come at the cost of no additional capacity as the Multicast Controller can terminate streams without viewers as it can discriminate between streams that are simply being cached opportunistically from streams which are actively being viewed.

9.3.4 Channel Change Performance

This section will illustrate several findings:

- Changing to an uncached multicast channel takes the same amount of time as changing to a unicast channel.
- Channel change time is not impacted by any potential delays related to acquiring multicast channel data or joining new multicast groups. If the channel is cached, there is no need to acquire channel map data or to join a multicast group, and if the channel is uncached, retrieval of initial segments will be via unicast.
- Time to multicast, however, can be impacted by channel change design decisions.
- Channel changing to a cached multicast channel takes roughly half the number of round-trip times as changing to an uncached channel.

9.3.4.1 Channel Change to Cached Channel

When the Gateway has dedicated a spare multicast receiver to receive and cache content, channel changes to a cached channel are quick and efficient. The following figure is a variation on Figure 3 and shows the sequence of events that occur when accessing a channel that may be cached.

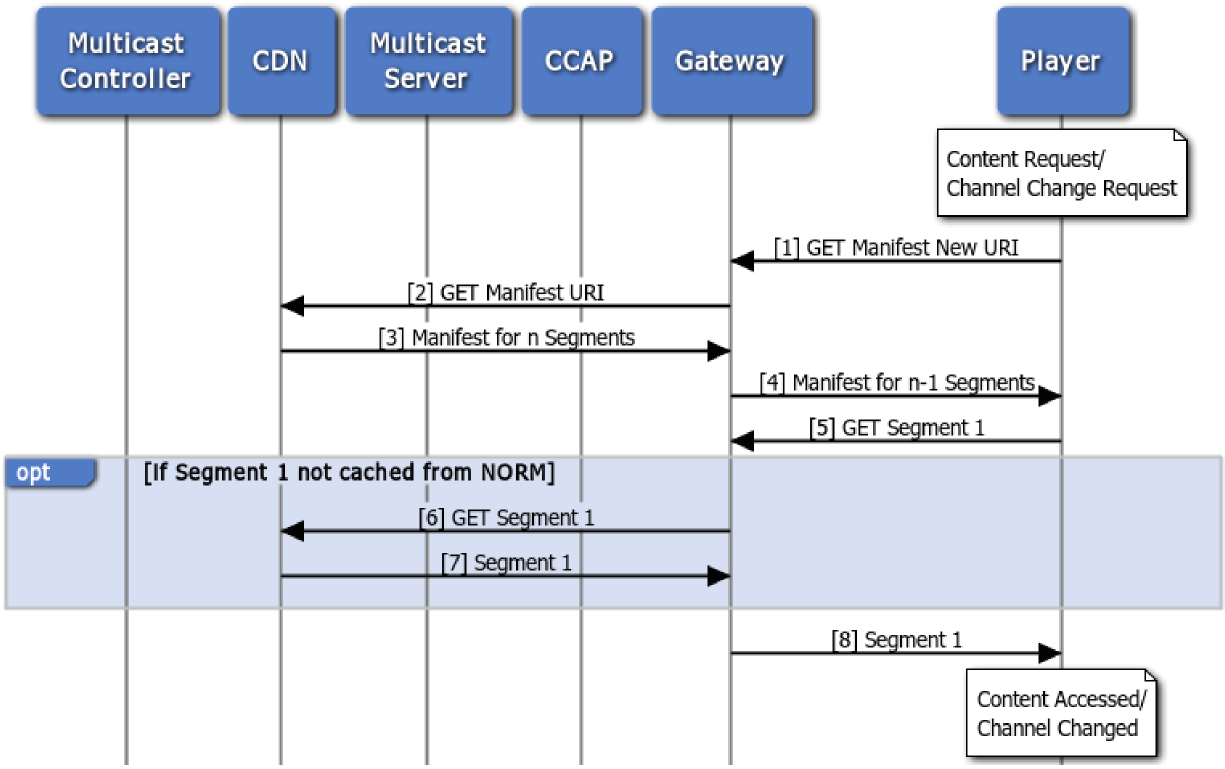

Figure 12 – Content Access Sequence for Cached/Uncached Content

This sequence shows that a full round-trip time to the CDN (steps 6 and 7) can be eliminated from the channel change time if a multicast receiver is being used in this way. Neglecting processing time in the Gateway, channel change time is driven largely by two exchanges: one higher latency exchange between the Player and the Origin/CDN for the Manifest and another negligible latency exchange from the Player to the Gateway for the cached video Segment file.[20]

9.3.4.2 Channel Change to Uncached Channels

When the Player requests a channel that is not already being cached by the Gateway, the likely steps are quite similar. In this case, steps 6 and 7 from Figure 12 are followed as the first segment file cannot be cached unless a spare multicast receiver has been allocated to this function.

It is important to note that are no multicast-related steps in this sequence and there is no dependency on any multicast steps succeeding for the channel change to become complete. Thus, the performance of an uncached M-ABR channel change is no different than the performance of a unicast ABR channel change. (Comparing Figure 3 to Figure 12 illustrates this fact.)

The performance difference between the cached and uncached channel change cases is largely the difference in the latency of the exchange between the Player and Gateway (steps 5 and 8 in the cached case) and the Player and Origin/CDN (steps 5, 6, 7 and 8 in the uncached case). Given that the Player and Gateway are on the same LAN segment, their latency is negligible, which means that the cached channel change is roughly twice as fast as the uncached channel change, thus illustrating the importance of caching channels as a Best Practice identified in Section 9.3.3.

[20] This also neglects an exchange for getting any keys needed for decoding the video segments in the manifest, but this time is identical in both the cached and the uncached case.

9.3.5 Time to Multicast Performance

The previous section pointed out that changing to a cached channel improves channel change performance over unicast by 50% and that changing to an uncached channel has essentially the same performance as unicast. Thus, from an end-user perspective, an M-ABR system is as good or better than a traditional ABR system.

However, for operators, there are other channel-change-related metrics that matter more from a network efficiency perspective than they do from an end-user QoE perspective. Chief among these is time to multicast - which is the amount of time from the Gateway receiving a request for a different video asset URI to the time the first multicast delivery of a segment from that asset.

9.3.5.1 Channel Change to Cached Channel

Much like channel change time, the time to multicast when changing to a cached channel is much better than when changing to an uncached channel. In fact, in the case of a cached channel, time to multicast is 0 as the new channel is already being received via multicast. This is a situation where the designed-in optimizations have been fully leveraged and the power of the channel caching Best Practice can be fully realized.

Given the QoE and efficiency benefits of caching channels, one can see why operators are researching algorithms to better predict future channel change behavior and thereby better predict which M-ABR channels should be cached.

9.3.5.2 Channel Change to Uncached Channels

When an unused multicast receiver is available for the new channel, the Gateway does not need to leave an existing multicast group before joining a new group. Thus, this scenario can lead to better time to multicast than when all receivers are busy and the Gateway needs to leave a multicast group.

The time it takes to complete the sequence shown earlier in Figure 5 entirely determines the time to multicast performance. However, there are several optional steps in this sequence which can impact the time to multicast.

Steps 2 and 3 are only necessary if the Gateway does not have *a priori* knowledge of the channel map. As discussed in 9.2.3, Gateways having advance knowledge of multicast channel map information is a best practice as it eliminates the need for these additional steps.

Steps 5 through 9 are only necessary if there are no free multicast receivers for the new channel. The recognition that these steps could be parallelized allows the identification of a new best practice.

Best Practice: Gateway should be designed such that it is possible to Join a new multicast group before the Leave for an existing multicast group is completed.

While time to multicast is important for efficiency, it should also be noted that there may be other system-level considerations that need to be taken into account when joining multicast groups. For example, initial CCAP implementations may have performance limitations related to the rate of IGMP messaging they can support. Thus, Gateway designers may want to consider delaying joining groups for multicast cache filling until the Player has been "tuned" to an M-ABR channel for some amount of time. In this way, channel surfing behavior can be handled largely via unicast means, and longer-term viewing enables multicast.

Design Consideration: Joining and leaving multicast groups consumes system resources; operators should examine the tradeoff between joining multicast groups as quickly as possible (which may increase multicast efficiency) and the CCAP resource cost of rapid joins/leaves due to "channel surfing" behavior.

One issue with M-ABR system design is that Player behavior can vary across implementations and is often outside operator control. Most Player implementations request a number of segments immediately at the start of a new stream to fill their buffers as quickly as possible ([Pantos-14] hints at 3 segments). Given this behavior and given the duration of a video segment (typically at least 2 seconds), it may be the case that a Player requests 6 seconds of video via unicast before multicast can reasonably be established (depending on how quickly the segments can be retrieved via unicast). This Player initial buffering behavior can provide a window for multicast delivery to begin and starting multicast before that window may not improve efficiency.

This may be best illustrated by an example. A Player requests a new content stream which is being broken into 6 second segments. The Player initially requests 3 segments in parallel which take 2 seconds to retrieve. Thus, the Player has 18 seconds of video buffered and retrieved via unicast in 2 seconds. To maintain a consistent buffer level the Player will typically request the 4th segment at the end of the first segment, or 6 seconds later. Since the initial 3 segments are retrieved in parallel, there is little benefit in having a time to multicast significantly less than 8 seconds as the first request outside the initial request burst does not occur until this time. Thus, for this type of parallel initial request behavior, the time to multicast only needs to be less than the time of the Player's first periodic, non-burst request.[21]

Design Consideration: Operators should consider Player behavior when attempting to optimize time to multicast. Unless time to multicast is low enough to fill part of a Player's initial burst of requests, Gateways may have additional time margin to acquire multicast content as it may be several seconds before the Player's first periodic request after its initial burst.

9.3.6 Time from Live

A third Key Performance Indicator for M-ABR service is the Time from Live. This is defined as the time difference, within the content stream itself, between the M-ABR stream and the version of the stream delivered via QAM. These time differences can create user dissatisfaction. For example, in current cable networks it is possible to have similar time discrepancies between the SD and HD versions of the same channel; thus, a subscriber with the same show on SD in the kitchen and on HD in the family room can hear the difference between the two streams.

Design Consideration: Operators should endeavor to minimize the time differences within different versions of the same stream.

Given that M-ABR is always at least one segment behind, ABR time from live cannot be completely brought to zero. However, limiting time from live to a single segment is a good design goal. Further, minimizing the segment duration will minimize the time from live.

Design Consideration: Shorter video segments generally lead to shorter time from live.

9.3.7 Emergency Alerts

Emergency alerts take many forms. Emergency alerts can be crawlers providing weather/AMBER alerts across the bottom of the screen. Less frequently, they can also be Emergency Action Notifications (EANs). In the case of an EAN, the entire video feed is pre-empted by a new video stream authorized by the President. As all viewers must be provided the EAN, it is clearly more efficient to deliver this content via multicast than via unicast.

However, this remains an area of open research. Beyond the clear benefit of multicasting some forms of emergency alerts (e.g., EAN), the best practices and design considerations for this technology remain open. Thus, the working group left this as an area for future work.

9.4 Operational Support Design Considerations

9.4.1 Viewership Reporting

To support M-ABR, a new IPDR/SP Service Definition has been developed by the working group. This Service Definition is intended to allow operators to have a complete picture of M-ABR group membership. The data includes the Join and Leave time for every (S,G) watched by the Gateways served by a given CCAP.

Best Practice: For viewership data on M-ABR content, the IP Multicast Stats Service Definition should be used.

It should be noted that this information could be useful when determining what content to multicast in the future and can be used to feed back data into a Policy-Driven Multicast content selection system.

9.4.2 Key Performance Indicators

Operational Support Systems are key to understanding the performance of a deployed M-ABR system. There are a number of KPIs which operators should consider monitoring to ensure that their M-ABR system is functioning as designed and to examine for discovering potential system performance improvements. This is an area of future work for the IP Multicast Focus Team and new management object models are likely to be developed to incorporate some of these KPIs.

[21] Clearly, with shorter duration segments this time is reduced, but operators may intentionally design for the 2nd or 3rd periodic request. Again, there is no QoE impact from increased time to multicast – only reduced efficiency.

As Channel Change Time can largely be captured by the Gateway it is possible to monitor this from the Gateway perspective. For example, the Gateway could start a timer each time a request for new managed video content is made and then terminate that timer when the corresponding response is complete.[22] Similarly, Time to Multicast could also be measured by utilizing a timer between that initial content request from the Player and the corresponding response arriving via multicast.

Time from live, however, is more of a system-level metric, which would be extremely difficult to measure on an individual Gateway. This might be possible to measure within the back office by somehow comparing unicast content streams from the CDN to multicast streams for the same content. However, even this would likely prove challenging as identifying the same internal point in two content streams relative to their delivery streams is difficult.

There are a number of additional KPIs which might give insight into other aspects of system performance. In particular, metrics related to cache hit rate and additional viewership data would likely prove useful to operators evaluating their multicast gain and potentially tuning their system.

While traditionally cache hit rate has been used in the web caching domain to look at how useful the cached content has been across the set of users utilizing a given cache, in the M-ABR model cache hit rate provides insight on the amount of content retrieved via unicast (a cache miss) versus the amount of content delivered via multicast (a cache hit). If all content uses the same segment size, then the percentage of multicast content consumption versus unicast content consumption could be monitored. As changes are made to the set of streams being multicast or other back-office algorithms, this metric could be utilized to evaluate the effectiveness of those changes.

As discussed earlier, continuing to receive an unwatched multicast channel which becomes a watched multicast channel can provide substantial improvements in channel change time and time to multicast. Thus, algorithms for identifying channels which should be cached are extremely valuable as the ones which best predict future channel change activity will likely provide the best QoE for subscribers. There are two potential classes of metrics to look at this aspect of system performance. One might count the number of segments which are received via multicast, but never requested from the cache (i.e., are never viewed). Presumably, implementations with fewer "cached, but not played" segments are utilizing their cache's more efficiently. As "cached, but not viewed" content is neither a cache hit or a cache miss, this provides a third caching-related metric which could prove to be valuable for operators.

Additionally, it should be noted that the "viewership reporting" described in the previous section makes no distinction between content which is actually being viewed versus content which is being cached, but not viewed. Thus, per (S,G), metrics on segments received, but not viewed can provide data from the Gateway which is not available on the CCAP (although this type of information may be provided to the Multicast Controller.)

In DOCSIS some of the most important physical layer statistics relate to corrected and uncorrectable codewords. As NORM also utilizes Reed-Solomon FEC, these same metrics - but at the multicast transport layer - can provide operators insight into the performance of their transport-layer FEC which can be used for tuning the amount of FEC protection required in different situations.

Currently there is no standard instrumentation for these or other KPIs which might be utilized to monitor and tune an M-ABR system, but, as mentioned previously, this is an item the working group is planning on addressing as part of their future work.

[22] Players also report a number of metrics such as Channel Change Time so adding this instrumentation to the Gateway may be unnecessary.

10 CONCLUSIONS

This technical report has discussed many aspects of Multicast-ABR video systems. It has defined a reference architecture and discussed the interfaces between the various components of that architecture. It has explored many aspects of an M-ABR deployment and identified a set of current best practices. Where significant tradeoffs exist, design considerations were also identified which help to crystallize the trade space architects should be evaluating.

Appendix I Example M-ABR Sequence Diagrams

This Appendix provides a number of sequence diagrams to illustrate the basic protocol exchanges which underpin Multicast-ABR. However, they are not based on any specific implementation and are optimized for clarity - not performance. In particular, a number of these diagrams show steps which could be optimized by parallelization, but for clarity they are shown sequentially.

I.1 Video-Related Sequences

I.1.1 Content Delivery with Multicast Cache Check

The following figure is identical to Figure 4, but also includes message details for each step in the exchange. These message details are not based on an actual implementation and should be used just for reference. They are intended to remove ambiguity from descriptions in the body of the technical report.

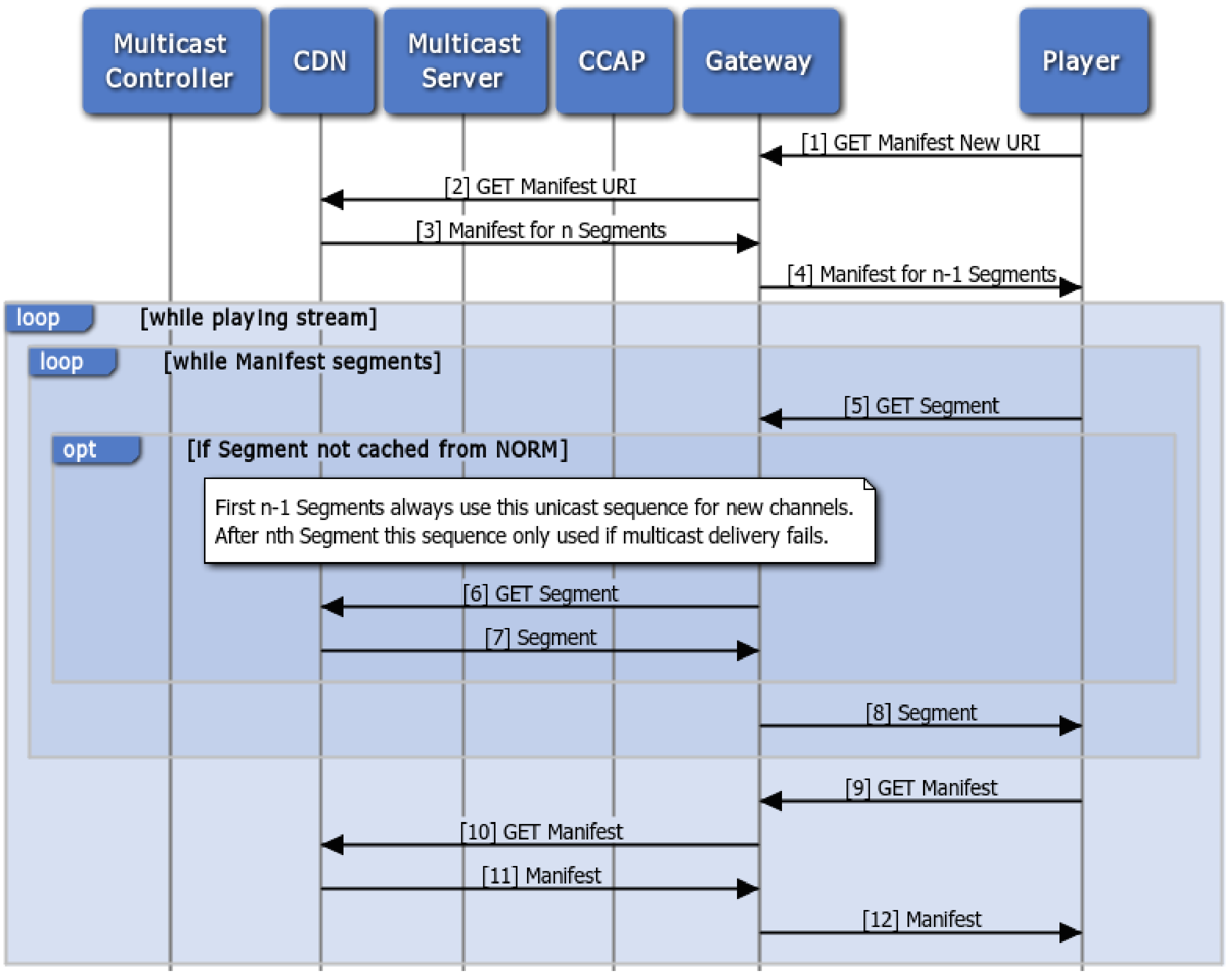

Figure 13 – Content Delivery with Multicast Cache Check

The corresponding messages are detailed in the following table.

Table 6 – Content Delivery with Multicast Cache Check

Step	Message Details
1	saddr=Player daddr=Origin GET http://devimages.apple.com/iphone/samples/bipbop/gear2/prog_index.m3u8 HTTP/1.1 Host: devimages.apple.com X-Playback-Session-Id: 137B15EC-BFFE-4E41-B95A-3480DFB99274 Proxy-Connection: keep-alive Accept: */* User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2 Referer: http://devimages.apple.com/iphone/samples/bipbopgear2.html Accept-Encoding: gzip Connection: keep-alive
2	saddr=Gateway daddr=Origin GET /iphone/samples/bipbop/gear2/prog_index.m3u8 HTTP/1.1 Host: devimages.apple.com X-Playback-Session-Id: 137B15EC-BFFE-4E41-B95A-3480DFB99274 Accept: */* User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2 Referer: http://devimages.apple.com/iphone/samples/bipbopgear2.html Accept-Encoding: gzip Connection: keep-alive
3	saddr=Origin daddr=Gateway HTTP/1.1 200 OK Server: Apache ETag: "50117c8233644c19b5ab49551b72507f:1239907416" Last-Modified: Thu, 16 Apr 2009 18:43:36 GMT Accept-Ranges: bytes Content-Length: 7019 Content-Type: audio/x-mpegurl Date: Wed, 03 Sep 2014 18:39:19 GMT Connection: keep-alive #EXTM3U #EXT-X-TARGETDURATION:10 #EXT-X-MEDIA-SEQUENCE:0 #EXTINF:10, no desc fileSequence0.ts #EXTINF:10, no desc fileSequence1.ts #EXTINF:10, no desc fileSequence2.ts #EXTINF:10, no desc fileSequence3.ts #EXTINF:10, no desc fileSequence4.ts #EXTINF:10, no desc fileSequence5.ts
4	saddr=Origin daddr=Player HTTP/1.1 200 OK Content-Length: 7019 ETag: "50117c8233644c19b5ab49551b72507f:1239907416" Date: Wed, 03 Sep 2014 18:39:19 GMT Last-Modified: Thu, 16 Apr 2009 18:43:36 GMT Server: Apache Accept-Ranges: bytes Content-Type: audio/x-mpegurl Connection: keep-alive #EXTM3U #EXT-X-TARGETDURATION:10

Step	Message Details
	#EXT-X-MEDIA-SEQUENCE:0 #EXTINF:10, no desc fileSequence0.ts #EXTINF:10, no desc fileSequence1.ts #EXTINF:10, no desc fileSequence2.ts #EXTINF:10, no desc fileSequence3.ts #EXTINF:10, no desc fileSequence4.ts #EXTINF:10, no desc fileSequence5.ts
5	saddr=Player daddr=Origin GET http://devimages.apple.com/iphone/samples/bipbop/gear2/fileSequence0.ts HTTP/1.1 Host: devimages.apple.com X-Playback-Session-Id: 137B15EC-BFFE-4E41-B95A-3480DFB99274 Proxy-Connection: keep-alive Accept: */* User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2 Referer: http://devimages.apple.com/iphone/samples/bipbopgear2.html Accept-Encoding: identity Connection: keep-alive
6	saddr=Gateway daddr=Origin GET /iphone/samples/bipbop/gear2/fileSequence0.ts HTTP/1.1 Host: devimages.apple.com X-Playback-Session-Id: 137B15EC-BFFE-4E41-B95A-3480DFB99274 Accept: */* User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2 Referer: http://devimages.apple.com/iphone/samples/bipbopgear2.html Accept-Encoding: identity Connection: keep-alive
7	saddr=Origin daddr=Gateway HTTP/1.1 200 OK Server: Apache ETag: "4611f4bcccb6f95f69041e6d48b058f9:1239907353" Last-Modified: Thu, 16 Apr 2009 18:42:33 GMT Accept-Ranges: bytes Content-Length: 414540 Content-Type: video/mp2t Date: Wed, 03 Sep 2014 18:39:19 GMT Connection: keep-alive {Binary Content Omitted}
8	saddr=Origin daddr=Player HTTP/1.1 200 OK Content-Length: 414540 ETag: "4611f4bcccb6f95f69041e6d48b058f9:1239907353" Date: Wed, 03 Sep 2014 18:39:19 GMT Last-Modified: Thu, 16 Apr 2009 18:42:33 GMT Server: Apache Accept-Ranges: bytes Content-Type: video/mp2t Connection: keep-alive {Binary Content Omitted}
9	Refer to step 1

Step	Message Details
10	Refer to step 2
11	Refer to step 3
12	Refer to step 4

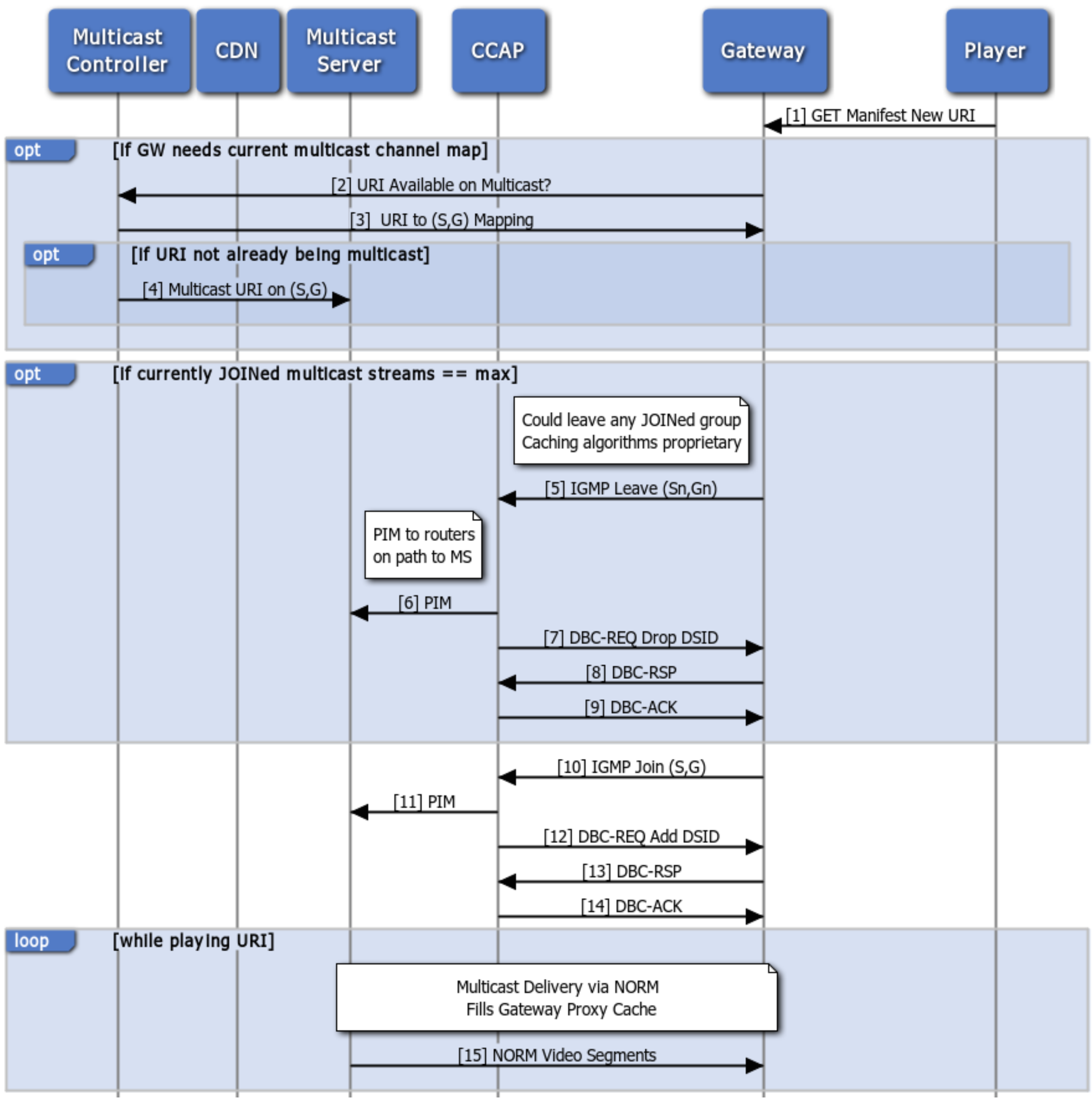

Figure 14 – Multicast Cache Filling

The corresponding messages are detailed in the following table.

Table 7 – Initial Request – Unvailable via Multicast

Step	Message Details
1	saddr=Player daddr=Origin GET http://devimages.apple.com/iphone/samples/bipbop/gear2/prog_index.m3u8 HTTP/1.1 Host: devimages.apple.com X-Playback-Session-Id: 137B15EC-BFFE-4E41-B95A-3480DFB99274 Proxy-Connection: keep-alive Accept: */* User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_4) AppleWebKit/537.78.2 (KHTML, like Gecko) Version/7.0.6 Safari/537.78.2 Referer: http://devimages.apple.com/iphone/samples/bipbopgear2.html Accept-Encoding: gzip Connection: keep-alive
2	Proprietary HTTP Request
3	Proprietary HTTP Response
4	Proprietary Request
5	saddr: Gateway daddr: 224.0.0.22 IGMP Version: 3 Type: Membership Report (0x22) Header checksum: 0xf8fb [correct] Num Group Records: 1 Group Record : 224.1.1.1 Change to Exclude Mode (4) Aux Data Len: 0 Num Src: 0 Multicast Address: 224.1.1.1 (224.1.1.1)
6	PIM details are TBS
7	saddr: CCAP MAC daddr: CM MAC Dynamic Bonding Change Request Transaction ID: 12345 Number of Fragments: 1 Fragment Sequence Number: 1 DSID Encodings DSID Value: 234 Downstream Service Identifier Action: Delete (2) Downstream Reseq. Encodings: Reseq. DSID Flag: 1 (DSID is a resequencing DSID) DS Channel ID Array: 0x0102030405060708 Multicast Encodings Client MAC Address Encodings Action: Delete (1) Client MAC Address: Gateway MAC Multicast CMIM: eRouter (0x40) Multicast GMAC Address(es): GMAC of Content Stream Key Sequence Number: 0x01 HMAC Digest: varies
8	saddr: CM MAC daddr: CCAP MAC Dynamic Bonding Change Response Transaction ID: 12345 Confirmation Code: okay / success (0) Key Sequence Number: 0x01 HMAC Digest: varies
9	saddr: CCAP MAC daddr: CM MAC Dynamic Bonding Change Acknowledgment Transaction ID: 12345

Step	Message Details
	Key Sequence Number: 0x01 HMAC Digest: varies
10	saddr: Gateway daddr: 224.0.0.22 IGMP Version: 3 Type: Membership Report (0x22) Header checksum: 0xf9fb [correct] Num Group Records: 1 Group Record : 224.1.1.1 Change to Include Mode (3) Aux Data Len: 0 Num Src: 0 Multicast Address: 224.1.1.1 (224.1.1.1)
11	PIM details are TBS

12	saddr: CCAP MAC daddr: CM MAC
	Dynamic Bonding Change Request
	Transaction ID: 12346
	Number of Fragments: 1
	Fragment Sequence Number: 1
	DSID Encodings
	DSID Value: 123
	Downstream Service Identifier Action: Add (0)
	Downstream Reseq. Encodings:
	    Reseq. DSID Flag: 1 (DSID is a resequencing DSID)
	    DS Channel ID Array: 0x0102030405060708
	Multicast Encodings
	    Client MAC Address Encodings
	        Action: Add (0)
	        Client MAC Address: Gateway MAC
	        Multicast CMIM: eRouter (0x40)
	        Multicast GMAC Address(es): GMAC of Content Stream
	Key Sequence Number: 0x01
	HMAC Digest: varies

13	saddr: CM MAC daddr: CCAP MAC Dynamic Bonding Change Response Transaction ID: 12346 Confirmation Code: okay / success (0) Key Sequence Number: 0x01 HMAC Digest: varies
14	saddr: CCAP MAC daddr: CM MAC Dynamic Bonding Change Acknowledgment Transaction ID: 12346 Key Sequence Number: 0x01 HMAC Digest: varies
15	See Section I.2 for additional detail on NORM messaging.

I.2 NORM-Related Sequences

The following sequence diagrams provide more detail on how NORM is used to deliver streams of M-ABR segments. Both sequences are assuming unicast repair so there are no NACKs from the Gateway Cache and no NORM repair traffic from the Multicast Server.

These sequence diagrams also separate the Gateway's Embedded Multicast Client function from its Caching Proxy function. These two functions typically operate asynchronously. The Embedded Multicast Client receives segments via NORM and "stuffs" the proxy's cache with them as shown in Figure 15. This provides the multicast functionality for the Gateway. The Caching Proxy function is a unicast function and is typically the function that interacts with the Player. This is shown in Figure 16, which is essentially providing an additional layer of detail not shown in

Figure 13, which abstracts away the notion that partial segments might be delivered and that the gaps in these segments can be filled using Range Requests to the CDN (as described in Section 9.2.2.2).

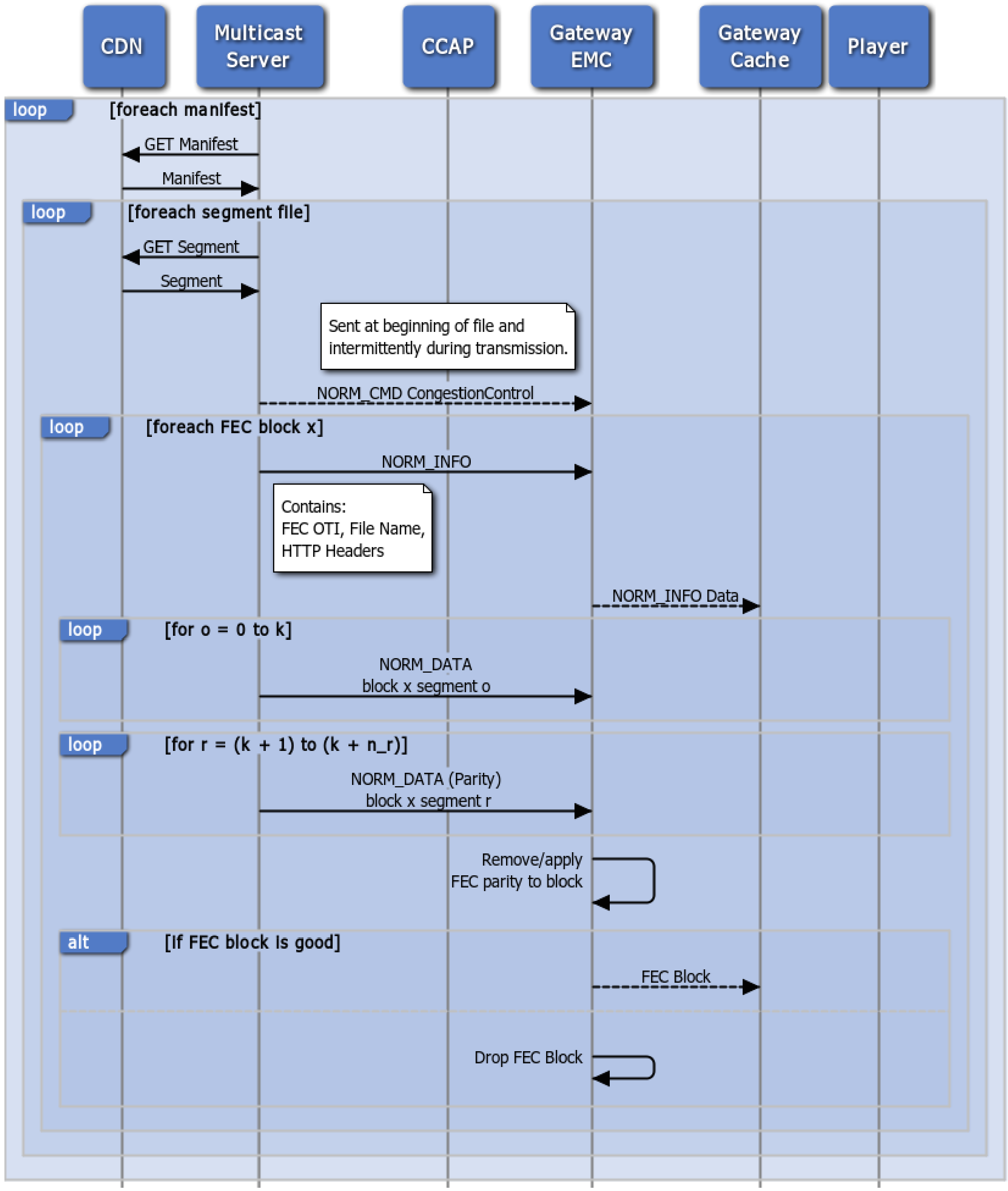

Figure 15 – NORM Delivery of Segments with Unicast Repair

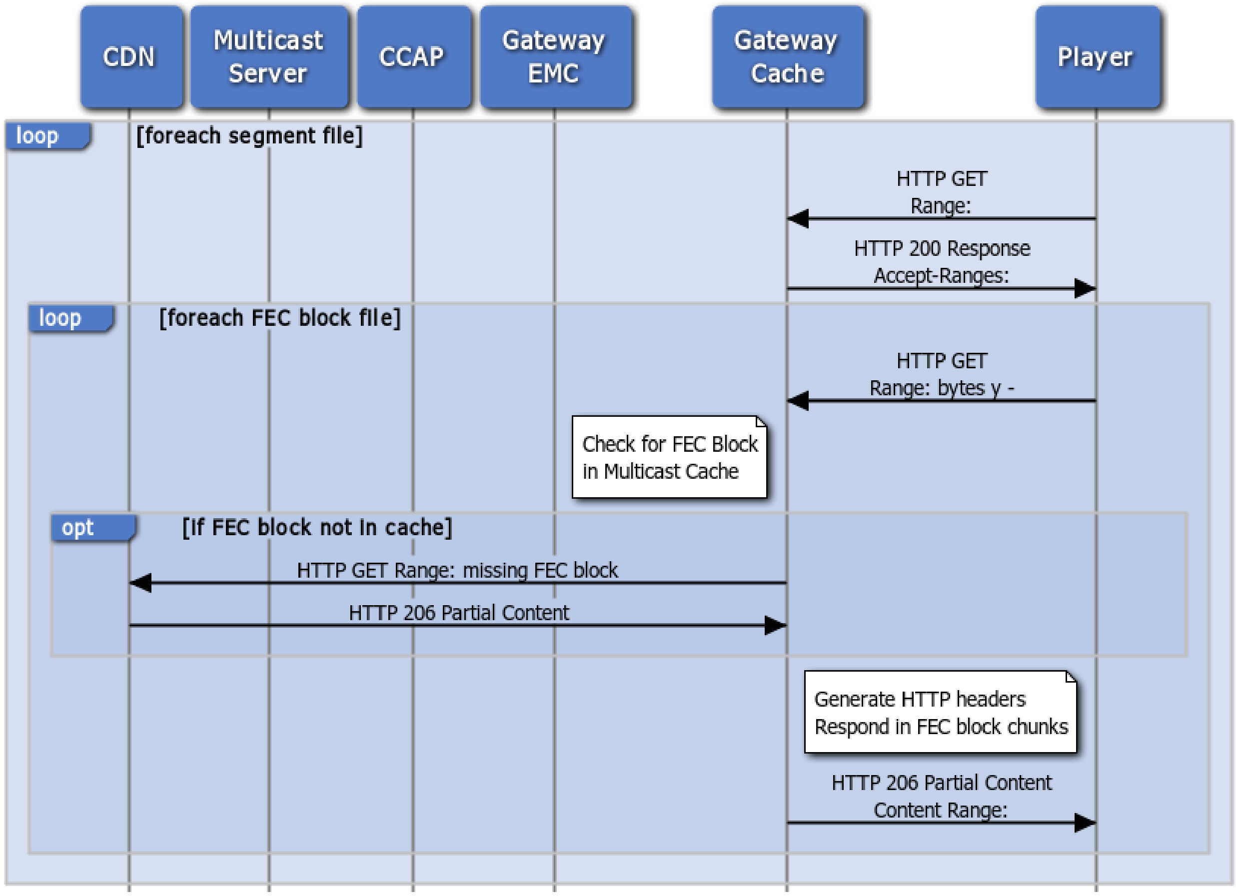

Figure 16 – NORM with Unicast Repair

As mentioned previously, Figure 16 should be considered in the context of Figure 13. This figure provides more detail on the way Range Requests can be used to fill in gaps in segment files that result from FEC blocks being dropped when errors occur.

Appendix II NORM_INFO Metadata Encoding

Each ABR video segment file is treated by NORM as a NormObject. Each NormObject can have associated metadata which can be delivered via NORM_INFO messaging. The payload_data portion of the NORM_INFO message is in an "application defined" format. This section defines a generic XML schema for NORM_INFO messages used for M-ABR systems.

This schema leverages a simple key-value pair mechanism similar to Apple's Plist format.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:ni="http://www.cablelabs.com/namespaces/multicast/NORM_INFO"
targetNamespace="http://www.cablelabs.com/namespaces/multicast/NORM_INFO"
elementFormDefault="unqualified" attributeFormDefault="unqualified" version="1.0">
  <xs:element name="metadata">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="key" type="xs:token"/>
        <xs:choice>
          <xs:element name="string" type="xs:string"/>
          <xs:element name="bool" type="xs:boolean"/>
          <xs:element name="integer" type="xs:int"/>
          <xs:element name="real" type="xs:double"/>
          <xs:element name="data" type="xs:base64Binary"/>
        </xs:choice>
      </xs:sequence>
      <xs:attribute name="version" type="xs:string" fixed="1.0"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

In addition to the schema, one standard key is defined - "HTTP_Headers". When used, the corresponding string value should contain all of the HTTP headers the Multicast Server wants associated with the corresponding video segment when the Gateway delivers it via unicast. The following is an example encoding of HTTP headers using this standard mechanism.

```
<?xml version="1.0" encoding="UTF-8"?>
<ni:metadata version="1.0"
xsi:schemaLocation="http://www.cablelabs.com/namespaces/multicast/NORM_INFO
norm_info.xsd" xmlns:ni="http://www.cablelabs.com/namespaces/multicast/NORM_INFO"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <key>HTTP_Headers</key>
  <string>HTTP/1.1 200 OK
Server: Apache
ETag: "17a9d47ae133f0bae8a0651fe97386d2:1239907490"
Last-Modified: Thu, 16 Apr 2009 18:44:50 GMT
Accept-Ranges: bytes
Content-Length: 923080
Content-Type: video/mp2t
Date: Wed, 27 Aug 2014 15:50:01 GMT
Connection: keep-alive</string>
</ni:metadata>
```

The following is a JSON representation of the same XML instance document.

```
{
  "ni:metadata": {
    "-xmlns:ni": "http://www.cablelabs.com/namespaces/multicast/NORM_INFO",
    "-xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
    "-version": "1.0",
    "-xsi:schemaLocation": "http://www.cablelabs.com/namespaces/multicast/NORM_INFO
norm_info.xsd",
    "key": "HTTP_Headers",
```

```
    "string": "HTTP/1.1 200 OK
Server: Apache
ETag: \"17a9d47ae133f0bae8a0651fe97386d2:1239907490\"
Last-Modified: Thu, 16 Apr 2009 18:44:50 GMT
Accept-Ranges: bytes
Content-Length: 923080
Content-Type: video/mp2t
Date: Wed, 27 Aug 2014 15:50:01 GMT
Connection: keep-alive"
  }
}
```

Appendix III Channel Map Reference Schema

This section describes a simple reference design of a schema intended to provide the Gateway with the basic ability to map Player content requests to Multicast Streams that contain the corresponding content. The Gateway can use this information to trigger a Join for a multicast group when such a group exists for the requested content.

Other related schemas from DVB and the OpenIPTV Forum exist, but these schemas don't specifically address the M-ABR (S,G) mapping issue that this schema is designed to cover. Further, the scope of these schemas is far larger than the scope of the M-ABR technical report; thus, these schemas contained additional features and complexity not needed to support this fundamental, but basic use case. That said, this schema is intended to be the minimum viable schema to perform this request to (S,G) mapping; actual implementations would likely define additional object and attributes.

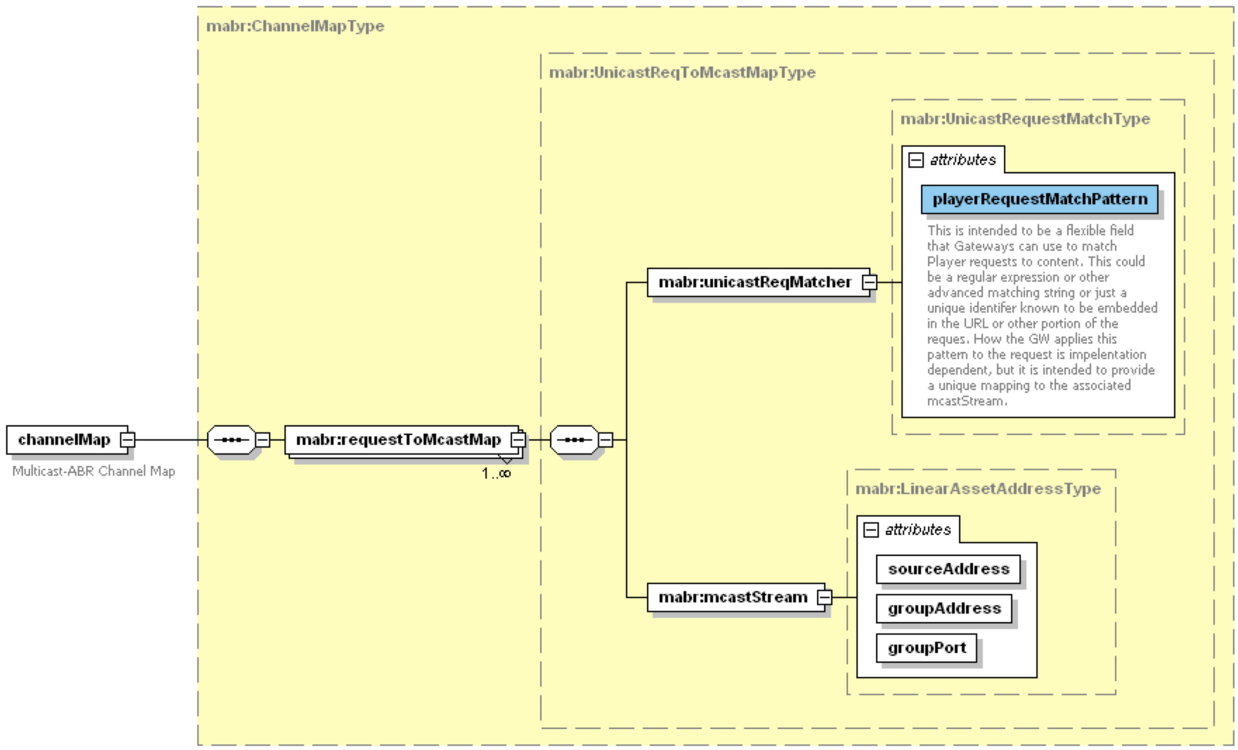

Figure 17 – Reference Channel Map Schema

III.1 LinearAssetAddressType Definition

Table 8 – LinearAssetAddressType Definition

Attribute Name	Type	Type Constraints	Units	Default Value
sourceAddress	string	dvb:IPOrDomainType		
groupAddress	string	dvb:IPOrDomainType		
groupPort	unsignedShort			

sourceAddress - the sourceAddress of the SSM (S,G) pair.

groupAddress - the groupAddress of the SSM (S,G) pair.

groupPort - the port number specific to the SSM group for the associated M-ABR stream.

III.2 UnicastRequestMatcherType Definition

Table 9 – UnicastRequestMatcherType Definition

Attribute Name	Type	Type Constraints	Units	Default Value
playerRequestMatchPattern	string			

playerRequestMatchPattern - This is intended to be a flexible field that Gateways can use to match Player requests for unicast ABR content to the same content's multicast stream (if available). This could be a regular expression or other advanced matching string or just a unique identifier known to be embedded in the URL or other portion of the request. How the GW applies this pattern to the request is implementation dependent, but it is intended to provide a unique mapping to the associated mcastStream.

III.3 UnicastRequestToMcastMapType Definition

This object contains a single object of type UnicastRequestMatcherType (its unicastReqMatcher) and a single object of type LinearAssetAddressType (its mcastStream).

III.4 ChannelMapType Definition

This object contains an unbounded sequence of objects of type UnicastRequestToMcastMapType in its unicastReqToMcastMap object.

III.5 Reference Channel Map Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:mabr="urn:com:cablelabs:mabr:2014-09-30"
  xmlns:dvb="urn:dvb:metadata:iptv:sdns:2008-1"
  targetNamespace="urn:com:cablelabs:mabr:2014-09-30" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import namespace="urn:dvb:metadata:iptv:sdns:2008-1"
schemaLocation="./sdns_v1.4r13.xsd"/>
  <xs:element name="channelMap" type="mabr:ChannelMapType">
    <xs:annotation>
      <xs:documentation>Multicast-ABR Channel Map</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="ChannelMapType">
    <xs:sequence>
      <xs:element name="requestToMcastMap" type="mabr:UnicastReqToMcastMapType"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="UnicastReqToMcastMapType">
    <xs:sequence>
      <xs:element name="unicastReqMatcher" type="mabr:UnicastRequestMatchType"/>
      <xs:element name="mcastStream" type="mabr:LinearAssetAddressType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="LinearAssetAddressType">
    <xs:attributeGroup ref="mabr:SSMAddressType"/>
  </xs:complexType>
  <xs:complexType name="UnicastRequestMatchType">
    <xs:attributeGroup ref="mabr:UnicastMatcherType"/>
  </xs:complexType>
  <xs:attributeGroup name="SSMAddressType">
    <xs:attribute name="sourceAddress" type="dvb:IPOrDomainType" use="required"/>
    <xs:attribute name="groupAddress" type="dvb:IPOrDomainType" use="required"/>
    <xs:attribute name="groupPort" type="xs:unsignedShort" use="required"/>
```

```
  </xs:attributeGroup>
  <xs:attributeGroup name="UnicastMatcherType">
    <xs:attribute name="playerRequestMatchPattern" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation>This is intended to be a flexible field that Gateways can
use to match Player requests to content. This could be a regular expression or other
advanced matching string or just a unique identifier known to be embedded in the URL
or other portion of the request. How the GW applies this pattern to the request is
implementation dependent, but it is intended to provide a unique mapping to the
associated mcastStream.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:attributeGroup>
</xs:schema>
```

Appendix IV Acknowledgements

This technical report was developed and influenced by numerous individuals representing many different vendors and organizations. CableLabs hereby wishes to thank everybody who participated directly or indirectly in this effort.

CableLabs wishes to recognize the following individuals for their significant involvement and contributions to the V01 technical report:

Dan Torbet – ARRIS

John Ulm – ARRIS

Ian Wheelock – ARRIS

Qin-Fan Zhu – Casa

Sangeeta Ramakrishnan – Cisco

John Bevilacqua – Comcast

Gene Granados – Cox Communications

Tom Gonder – Time Warner Cable

CableLabs technical reports acknowledge individuals who contributed to the development of that document. Their work has contributed to this technical report as well.

Matt White and Greg White – CableLabs

Andrew Sundelin – Consultant to CableLabs