EFFECTIVE UTILIZATION OF M-ABR (MULTICAST ASSISTED ABR) USING BIG DATA AND REAL-TIME ANALYTICS

Sridhar Kunisetty, Jeffrey Tyre, and Robert Myers
ARRIS

Abstract

Adaptive Bit Rate (ABR) streaming protocols support multi-screen devices like tablets, smart phones, gaming devices, and smart TVs for accessing Over-The-Top (OTT) video content. ABR streaming is now making its way to the primary TV screen in the home in the form of next generation IP set-top boxes (STBs), and with this a tremendous growth in ABR generated, HTTP unicast traffic is expected due to higher video resolutions and viewership concurrency as compared to second screen devices. The introduction of Multicast-assisted Adaptive Bit Rate (M-ABR) solutions is intended to help offset the potential impact of this unicast traffic growth by leveraging the bandwidth efficiencies of IP multicast over the operator network, while still enabling all pay TV subscriber devices in the home to support Adaptive Bit Rate streaming.

Actionable insight derived from collecting, processing, and analyzing the huge amount of data from the millions of home gateways provides several levels of valuable information that allows us to better manage the network by understanding the operational issues. Harnessing Big Data technologies and doing this in real-time, provides a feedback loop using which things can be changed dynamically at a granular level -- at a specific time of the day in a specific geographical area -- to better utilize the bandwidth and reduce operational complexity.

This paper provides an overview into how Big Data and Real-time Analytics enables insight into video consumption and network operational aspects to make M-ABR more effective.

M-ABR SOLUTION SUMMARY

Adaptive Bit Rate (ABR) protocols have become the mainstay of multi-screen devices like tablets, smart phones, gaming devices and smart TVs for accessing Over-The-Top (OTT) video content. Because of their explosive popularity, it is highly desirable for an operator to provide existing services to these IP Video, multi-screen devices. The ABR protocols have been optimized to deliver a wide range of bit rates for varying screen sizes from thumbnails to 4K Ultra HD, operating over IP network connections with fluctuating bandwidth and service levels.

Adaptive Bit Rate streaming is already used today by pay TV operators to deliver linear content over IP in the home to second-screen subscriber devices such as PCs, laptops, tablets, and mobile phones. ABR streaming is now making its way to the primary TV screen in the home in the form of next generation IP STBs. The intent of the M-ABR solution is to enable all pay TV subscriber devices in the home to be ABR-based using unicast HTTP, and to leverage the bandwidth efficiencies of IP multicast for linear channel delivery across the operator network. The economic benefits of the M-ABR solution have been previously documented[2].

The following diagram illustrates the major components of a Multicast-assisted ABR solution based on recently released CableLabs specifications. The CableLabs M-ABR specification[1] uses the IEFT RFC-5740's NORM (NACK-Oriented Reliable Multicast) transport protocol to propagate multicast streams over the operator network.

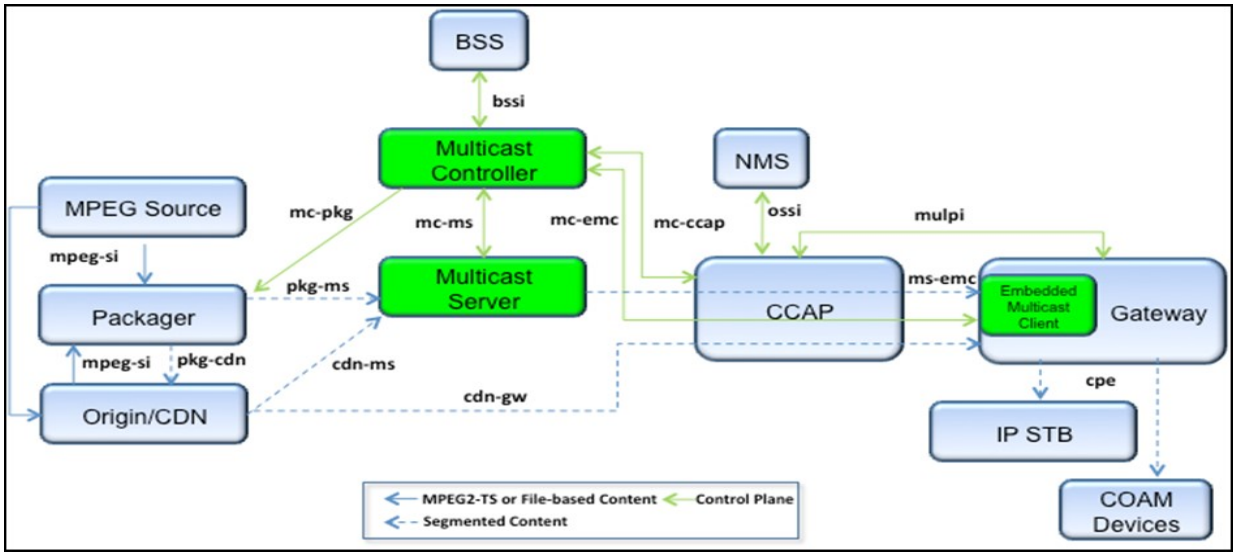

Figure 1: Major components in a M-ABR System

In this pay TV operator solution, an *ABR transcoding-packaging system* publishes ABR content into the operator managed content delivery network (CDN). The *M-ABR Server* pulls this content from the CDN as a normal ABR client would do, and sends the content as a multicast transport over an IP routed network to a *M-ABR Edge Device* with an *Embedded Multicast Client (EMC).* The M-ABR Edge Device is often a home gateway representing the operator network termination point for the customer premise equipment (CPE). The addition of a *M-ABR Configuration Manager* may be deployed to support the setup and configuration details of the M-ABR Servers and Edge Devices with EMCs hosted on them.

The *M-ABR Controller* is responsible for defining the mapping of specific ABR delivered streams for the M-ABR Servers to multicast delivery of the ABR content and provide mappings to edge devices so they know what services are available and how to connect to them. Management of this mapping can be either operator scheduled or dynamic based on tracking ABR client channel viewership trends across service groups, as well as at regional and national levels.

The M-ABR Controller determines what channels are multicast by instructing M-ABR Servers to setup and subsequently tear down multicast streams as ABR stream viewership changes or based on operator scheduled multicast events. Having a single management platform for both scheduled and dynamic viewership allows optimization of network, M-ABR Server, and edge device cache resources.

The M-ABR Embedded Multicast Client (EMC) – Window to the New Age of IP Video Analytics

The Embedded Multicast Client (EMC) is an application running in the M-ABR Edge Device that manages all aspects of the NORM multicast inputs received from M-ABR Server and their conversion to HTTP unicast ABR streams delivered over a local network (e.g. subscriber home network) to ABR clients for playout. In an effort to standardize its implementation, an EMC application is currently being incorporated into the

Reference Design Kit for Broadband services (RDK-B) library.

EMC communicates with the M-ABR Configuration Manager which provides configuration information to EMC. This includes the M-ABR Controller URL to send heartbeats and receive a multicast channel map and other configuration parameters required for operation.

The following describes the process that occurs when an ABR player client makes a request for content through the M-ABR Edge Device with EMC running as an application. When EMC receives an ABR video segment requests from ABR player clients which have been proxied to it:

1. EMC checks if the ABR client URL request matches a stream that is currently being multicast and in the Edge Device's local cache. If not, EMC requests the video segment via HTTP unicast and returns it to ABR client. In this capacity, EMC acts as a traditional CDN transparent web cache.

2. If the video segment requested by the ABR client is present in the local cache, EMC returns it to the ABR client immediately. A video segment could already be present in the local cache due to a previous ABR client request or based on a scheduled operator request to pre-cache content based on projected program viewership such as for live sports events.

3. If the NORM multicast stream has not been joined, the EMC joins the multicast stream to receive video segments via multicast, then converts these from NORM to an ABR format (e.g. HLS, DASH ISO), and stores these as ABR segments in the Edge Device's local cache.

4. All subsequent ABR segment requests are delivered from the Edge Device cache based on step #1 above.

5. EMC periodically checks for updates to the configuration and multicast stream channel map.

Subscriber devices are HTTP unicast, ABR clients, whether they are IP STBs or other subscriber devices such as iPads, PCs, and smart phones. The fact that some of the ABR content is being carried over the operator network as multicast traffic is completely transparent to them. End ABR clients could themselves support the M-ABR processing. However, it is generally expected that this functionality to be the most prevalently implemented in home gateways to minimize CPE costs by deploying the M-ABR functionality in a single home device.

EMC periodically sends statistics to the Analytics System about the ABR stream sessions and other details, including what ABR streams are requested, the cache hit/miss ratio for multicast streams, multicast data reliability, etc. This data will be the backbone of the operational and subscriber viewership analytics driving which streams should be available via multicast.

With the introduction of the M-ABR and EMC-enabled home devices, a new window to pay TV operator IP Video-supported services will emerge. Similar to current legacy MPEG video delivery technologies, new tools will need to be developed in order to effectively manage the associated new service deployments. This paper explores the analytics potentially derived from collecting and processing the huge amount of data generated from millions of EMC-enabled home devices that could be instrumental in ensuring the successful deployment of Multicast-assisted ABR solutions.

OVERVIEW OF BIG DATA & REAL-TIME ANALYTICS

The advent of the Internet and the resulting data that needs to be managed, stored, and analyzed, over cheap commodity hardware that can easily scale horizontally, led to the emergence of modern day Big Data technologies. Terms typically used to characterize Big Data are volume, velocity, and variety[3]. Volume refers to the sheer amount of data that is being created continuously. Velocity is the speed with which this data can be processed and analyzed for a meaningful use. Variety is the different types and formats of data that is collected.

The core tenets of any Big Data system are its ability to collect a variety of large amounts of data, use parallel processing to analyze and process that information to derive meaningful insight, and to provide access to the derived information through dashboards, reports, and APIs[4]. All this can be achieved at near linear scalability - as the data size increases, one can throw more hardware at it and see processing complete in the same amount of time – and at the same time being tolerant to failures on any of the computer nodes processing that data.

Over the years, many companies, universities, and open source communities contributed to Big Data related technologies such as Hadoop [5], MapReduce[6], YARN[7], SpringXD[8], Spark[9], etc. While Hadoop is good at handling Big Data, it primarily deals with batch-oriented data processing and storage. Traditional Hadoop technologies are not well suited for real-time data processing needs. In recent years, the open source Spark framework has become the defacto choice for processing and analyzing big data in real-time.

While Spark's streamlined API allows representing complex pipelines in a few lines of code making distributed programming easier, its in-memory processing makes it an order of times faster than traditional Hadoop technologies[9][10]. Spark also makes the life of a data scientist easier by providing APIs for statistics, machine-learning, graph processing, etc. – all in a single framework while providing linear scalability.

Spark provides strong integration with Hadoop ecosystem and in many cases both of them are used together – Spark for real-time data processing & Hadoop for batch processing and to store the data.

In this paper, our primary focus is on real-time analytics and not batch processing of Big Data.

Applicability to M-ABR

A well-instrumented M-ABR eco-system mirrors the web world in terms of volume, velocity, and variety of data that needs the processing power of a Big Data system. A typical operator has a few million subscriber homes, each of them having a home gateway (containing EMC), along with one or more STBs and second-screen devices connected to the home gateway. Each of these millions of home-based subscriber devices are constantly producing events i.e., we have a system that is generating millions of data events every minute. In addition to the devices in the subscriber's home, the components in the content delivery side, such as Multicast Server, also produce huge amount of useful data.

Analyzing this data provides insights into the M-ABR ecosystem for understanding operational issues and generating a feedback loop to help in the automation for fixing the operational issues that were detected. In order for this to be effective, the collected data need to be analyzed in real-time. For example, if we detect that the network latency is beyond a particular threshold, then this information can be used for corrective actions immediately. Similarly, if the cache hit ratio is below the

99% watermark, then it tells you, in real-time, that there are some problems in the network that need immediate attention.

While, addressing operational issues is the main focus of this paper, the M-ABR Analytics system can also gain insight into quantifying user behavior to facilitate various types of analysis and recommendation systems. Client-facing applications resident on a EMC device or a web-application can be instrumented to collect user interaction and content consumption information to derive insight into user behavior with respect to video consumption. And when combined into external data such as EPG metadata & demographics, we can get an even deeper insight into user behavior. This paper does not elaborate on the user behavior analytics aspect.

ANALYTICS - DERIVING INSIGHT

Analytics System Overview

The Analytics System is broadly classified into three different layers:

- *Data Ingest Layer*: This layer facilitates collection of multi-formatted information from a variety of data sources in real-time, doing parsing and pre-processing as needed and passing it to the Analysis & Processing Layer. Optionally, a copy of the raw data is stored in Hadoop for long-term storage needs.

- *Analysis & Processing Layer*: This is where all the real-time processing and analytics happen. It consists of different processes all running asynchronously in parallel. One set of processes operate on the raw data, process them and stores in a No-SQL database. The other set of processes perform the next level of processing by doing summarizations, correlations, deeper analysis, etc.

- *Data Output Layer*: The resultant data is exposed through multiple methods – dashboards and charts that provide a visual view into the different operational and usage aspects of M-ABR and REST APIs that can be programmatically consumed by various services to act on the analyzed data.

A Spark cluster, running on multiple nodes, provides the Analysis & Processing Layer the ability to horizontally scale as the data traffic increases.

Goals of Real-time Analysis

The objective of the Analytics System is to benefit the operator by providing meaningful information to understand and address operational issues in the M-ABR eco-system.

High-level goals include:

- Understanding usage of M-ABR components, services, and network

- Understanding and classifying network and system limitations by analyzing trends of system usage with respect to scalability, capacity planning, and bandwidth management

- Creating dashboards to provide business intelligence, and APIs for downstream usage

- Answering key questions about the performance of multicast over DOCSIS, including the telemetry of any packet loss detected by the EMC

- Evaluating the workload within the EMC to perform the functions of the “transparent proxy cache”

- Examining the performance of the CMTS/CCAP for multicast delivery and implications related to its support for IGMP.

At a more detailed level, the Analytics Systems helps in analyzing and answering questions such as:

- How often does a file segment experience any packet loss?
- Is the packet loss characterized by factors such as time of day, downstream capacity, CMTS type/configuration, aspects of the HFC access network, IGMP traffic management, etc.?
- How does channel surfing affect the IGMP behavior of joins and leaves and when does it become more of a burden and less of a benefit?
- How quickly can initial tuning to a new channel result in moving from unicast video segments to multicast (cached) video segments?
- How long should the EMC remain on a multicast stream before it determines there is no benefit to caching stream segments?
- What is the timing of a cache segment to client request?
- What is the behavior of client change in quality using alternate, non-multicast variant playlists?

EXAMPLE DATA EVENTS AND INSIGHTS

Data Events

The Analytics System captures a variety of information. Some of them are:

- Segment data
- Multicast Session data
- Client Session data

Session Data

For each multicast video segment received on the EMC, this data event is produced containing attributes such as multicast service name, receive timestamp, missing packet count, client request timestamp, etc.

This data helps in analyzing the following:

- Segment packet loss details and whether the packet loss was sequential or multiple packet-loss across the segment
- Segment timing details such as, when it is received with and without error and how soon after the client requested the video segment
- Segment details about how long it is maintained in the EMC cache

Multicast Session Data

For each IGMP session on the EMC this data event is produced to track the lifecycle of the multicast stream. This data event contains attributes such as multicast service name, multicast join & leave timestamps, first segment received timestamp, total segment count, segment errors count, etc.

This data helps in analyzing the following:

- IGMP behavior on the EMC and the success of each multicast video segment received
- Estimate of the IGMP load on the CMTS as a viewer transitions from service to service
- Details on the multicast session, such as when it is initiated, when it receives data, when the first segment starts, how many segments had an error, etc.

Client Session Data

For each client HLS session associated to a multicast service, this data event is generated containing attributes such as multicast service name, start manifest timestamp, total segment count, initial unicast segment count, user agent, etc.

This data helps in analyzing the following:

- Whether the video segment was provided from the cache or required the EMC to pass the request through unicast on the DOCSIS network
- Estimate the initial unicast load for the client buffer fill as well as how often the client request a segment that results in a cache miss and goes unicast
- The experience of a client session using specific multicast service

Insights

Information that is collected and processed must provide actionable value to the operator. Most of the insights can be categorized into the following keys areas:

- Multicast Channel Usage / Channel Popularity
- Initial Unicast Traffic & Multicast Cache efficiency
- Network Behavior & Packet Errors
- EMC Client Interaction Behavior

This section shows a few example charts providing meaningful insights.

Channel Utilization

Figure 2 shows the number of client sessions that exist on a per multicast channel (stream) basis over a specified time period, termed as the Popularity of a Channel.

This information is helpful in environments where the number of channels that can be simultaneously multicast is limited due to bandwidth capacity limitations. It can help to determine which channels are more popular and when.

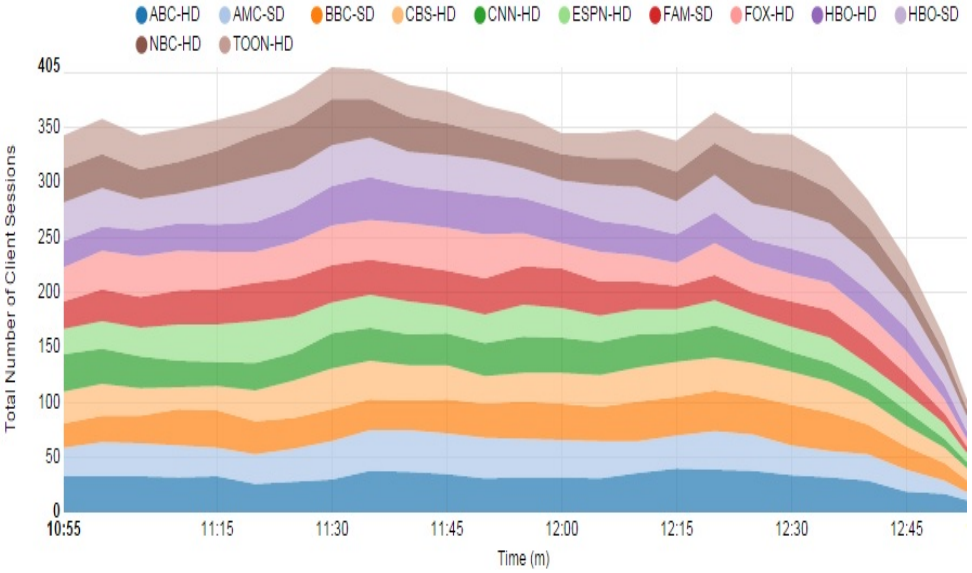

Figure 2: Channel Utilization

Missed Segments

Figure 3 shows the peak and average number of segments that could not be delivered from the EMC cache to a client device after the EMC has started receiving the multicast stream for that client session. In an optimal network this should be zero. Missed segments but can be impacted by issues in such as network errors or multicast server failures.

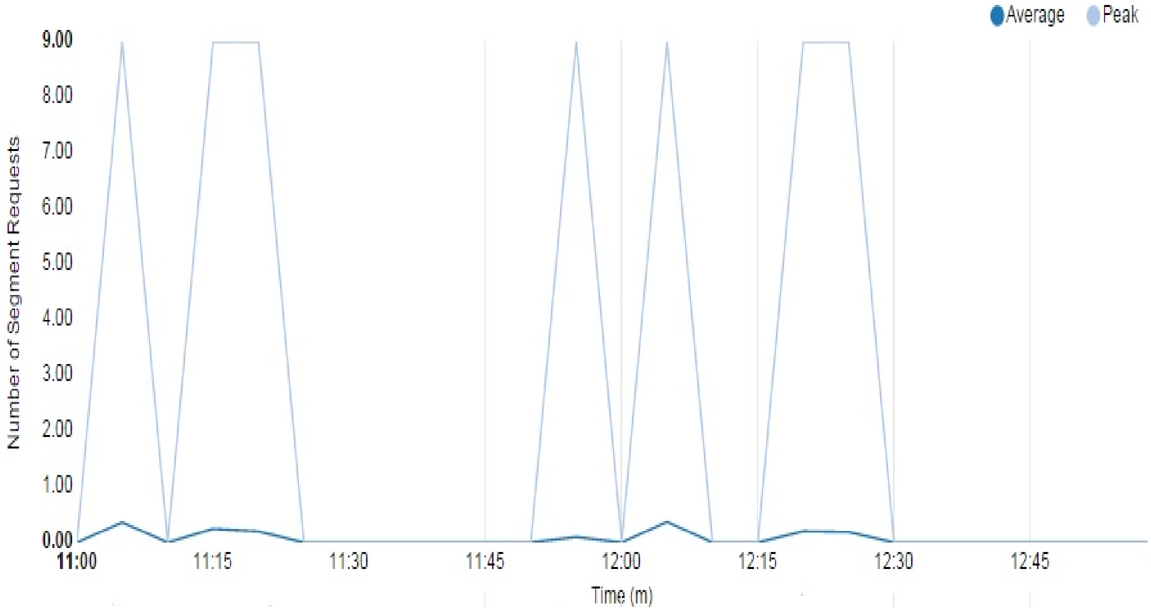

Figure 3: Missed Segments

Network Latency

Figure 4 shows the peak and the average time that it took for the EMCs to receive a segment from the multicast stream. This value is dependent on other factors such as duration of the segment, the available bandwidth and the multicast rate percentage. Optimally, the average value should be less than the segment duration to ensure that the EMC will always have available segments in its cache. If this value becomes greater than the segment duration it could indicate an issue somewhere in the network or in the multicast server.

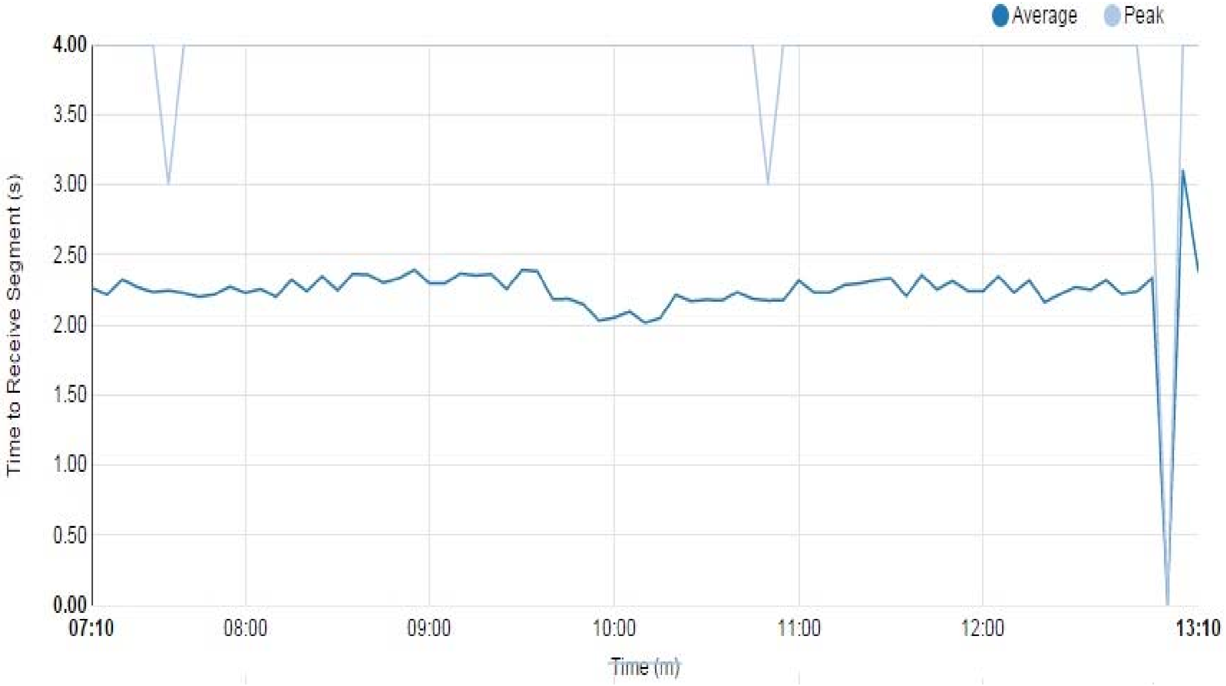

Figure 4: Network Latency

CONCLUSION

Multicast-assisted Adaptive Bit Rate (M-ABR) is a promising solution for operators as they roll out IP Video to Multiscreen devices since it helps in leveraging the bandwidth efficiencies of IP multicast for linear channel delivery over the operator network, while still enabling all TV subscriber home devices to support ABR streaming. Understanding the operational issues becomes essential for this new technology so that we can address those issues to improve efficiency, reduce cost and provide better user experience.

Actionable insight derived from collecting, processing, and analyzing data events provides several levels of valuable information. The volume, variety and speed of data generated from the M-ABR eco-system needs a Big Data and Real-time Analytics system.

In this paper, we provided an overview of the Big Data and Real-time Analytics technologies, discussed the sources of events in the M-ABR system, and provided examples of meaningful conclusions that can be ascertained from analyzed events useful in understanding and driving operational efficiencies.

ACKNOWLEDGEMENTS

This paper represents work that is being done in ARRIS Network & Cloud business unit.

REFERENCES

1. "IP Multicast Adaptive Bit Rate Architecture Technical Report", Cable Labs document OC-TR-IP-MULTI-ARCH-V01-141112, Nov 2014

2. Ulm, J., "Challenges delivering Multiscreen Linear TV Services: Multicast assisted ABR to the Rescue", SCTE Cable---Tec Expo, Fall 2014

3. Couts, A., "Breaking down 'big data' and Internet in the age of variety, volume, and velocity", Retrieved from

http://www.digitaltrends.com/web/state-of-the-web-big-data/, Sept 2012

4. Gandhi, B., Martinez-Smith, A., Kuhlman, D., “TV INSIGHTS - Application of Big Data to Television”, IBC 2015.

5. “Apache Hadoop”, Retrieved from http://hadoop.apache.org, Feb 2016.

6. "MapReduce Tutorial", Retrieved from http://hadoop.apache.org/docs/r2.7.2/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html, Jan 2016.

7. "MapReduce NextGen aka YARN aka MRv2", Retrieved from http://hadoop.apache.org/docs/r2.7.2/hadoop-yarn/hadoop-yarn-site/index.html, Jan 2016.

8. "Spring XD Guide", Retrieved from http://docs.spring.io/spring-xd/docs/1.3.1.RELEASE/reference/html/, Feb 2016.

9. “Spark, Lightning-fast cluster computing” Retrieved from https://spark.apache.org, Mar 2016.

10. “Spark the fastest open source engine for sorting a petabyte”, Retrieved from https://databricks.com/blog/2014/10/10/spark-petabyte-sort.html, Oct 2014.