ROUTE/DASH IP Streaming-Based System for Delivery of Broadcast, Broadband, and Hybrid Services

Gordon Kent Walker, Thomas Stockhammer, *Senior Member, IEEE*, Giridhar Mandyam, *Senior Member, IEEE*, Ye-Kui Wang, and Charles Lo

***Abstract*—The delivery of digital television services over broadcast channels using the MPEG-2 transport stream (TS) is a well-established concept. Despite the success of this technology, the advances and success of Internet protocol (IP)-based delivery models with browser-centric media endpoints that combine broadband and broadcast technologies as well as new service requirements have resulted in alternatives to the MPEG-2 TS. These IP-based approaches are considered essential to maintaining the success of broadcast-centric television services. This paper discusses an IP centric approach which is respectful of clean layering and combines the over the top and the broadcast worlds. Central to this approach is a protocol named real-time object delivery over unidirectional transport (ROUTE), which is based IETF protocols such as layered coding transport over user datagram protocol. The dynamic adaptive streaming over HTTP format is used for both broadband and broadcast delivery. HTTP is used as the broadband delivery protocol, and ROUTE is used as the broadcast delivery protocol. It is demonstrated that the ROUTE-based approach is a lean and powerful media delivery method optimized for streaming media and non-real time media delivery with available application layer forward error correction. Thanks to its merits, this approach has been incorporated into the advanced television systems committee 3.0 design.**

***Index Terms*—ALC, AL-FEC, CA, DRM, EME, FEC, HTML, HTTP streaming, IETF, LCT, MPEG DASH, MPEG-2 TS, MSE, ROUTE.**

I. INTRODUCTION

DIGITAL TeleVision (DTV) services over broadcast channels are a well-established concept. DTV services have been deployed since the 1990s utilizing MPEG-2 Transport Stream (TS) [1]. This broadly accepted and successful method is predicated on a certain set of assumptions (see Section II) that are appropriate to broadcast-only services.

HTTP adaptive streaming is currently used to deliver essentially all broadband streaming IP content. The different HTTP adaptive streaming technologies are converging to MPEG Dynamic Adaptive Streaming over HTTP (DASH) [2]. DASH format content delivered via HTTP is a very popular deployment choice. The widespread adoption of Internet streaming technologies has created opportunities for the convergence of Internet streaming and DTV delivery in a manner that enables the broadest possible service set.

Manuscript received August 14, 2015; revised December 28, 2015; accepted January 4, 2016. Date of publication February 11, 2016; date of current version March 2, 2016.

G. K. Walker, G. Mandyam, Y.-K. Wang, and C. Lo are with Qualcomm Inc., San Diego, CA 92121 USA (e-mail: gwalker@qti.qualcomm.com).

T. Stockhammer is with the Munich University of Technology, Munich 80333, Germany.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBC.2016.2515539

However, while HTTP over TCP is very good for broadband/unicast delivery, it is not appropriate as an end-to-end delivery protocol for broadcast. To best enable a complete system of hybrid broadcast and broadband services, an enhanced broadcast transport method named Real-time Object delivery over Unidirectional Transport (ROUTE) / DASH [3] has been developed for delivery of DASH-formatted content and Non-Real Time (NRT) data through broadcast channels. ROUTE/DASH is a broadcast object and media aware byte range delivery protocol based on MPEG DASH and Layered Coding Transport (LCT) [4] over User Datagram Protocol [5], with the use of Application Layer Forward Error Correction (AL-FEC), such as RaptorQ [6]. ROUTE/DASH is a logical extension of FLUTE/DASH as practiced in eMBMS. FLUTE while efficient for single objects is notably less flexible when it comes to delivering multiple object streams concurrently.

ROUTE provides general broadcast delivery capabilities similar to the broadband delivery capabilities provided by HTTP. In addition to supporting the file-based streaming techniques used in DASH, ROUTE provides media-aware content delivery which enables faster channel change. ROUTE/DASH works with any media codec, including scalable media codecs provided that the appropriate codec specific file format is specified.

The remainder of this paper is organized as follows. In Sections II–V, some background is provided, including key assumptions of MPEG-2 TS based DTV, the Internet streaming environment, the evolution of DTV broadcast physical layers, and the convergence of DTV and streaming IP. Section VI provides a detailed introduction of the ROUTE object delivery method, followed by a discussion of synchronization, system buffer model, use with AL-FEC, signaling, and integration with web-based applications in Sections VII–X.

The key use cases and how these are addressed by ROUTE/DASH are discussed. Clean layering and the reuse of existing technologies are emphasized.

II. Background on MPEG-2 TS Based DTV

A key innovation of MPEG-2 TS was the use of packet transport. The use of a one hundred eighty eight byte packet structure allowed the distribution of multiple streams of media in a common multiplex. This packet multiplex structure enabled so called statistical multiplexing, which in turn enabled gains in efficiency as compared to fixed bit rate allocation [7].

Upon the definition of MPEG-2 TS, this packet structure was incorporated into the structure of the first generation DTV physical layers for example ATSC [8] and DVB-T [9]. The operation of such MPEG-2 TS based systems is predicated on a number of key assumptions.

The principle among these assumptions is a constant delay across the delivery link. This is quite appropriate for typical satellite and terrestrial systems, which often have line of sight or near line of sight connections. Given this core assumption it was straightforward to deliver the time base on a per stream basis via the so called Packet Clock Reference (PCR).

Another key assumption is packet stream encryption. The individual packet streams in general carry one or more related media streams. The streams could be encrypted on a per packet stream basis to provide so called Conditional Access (CA).

The signaling structures for MPEG-2 TS were developed based on a series of binary tables. These binary table based signaling structures are compact, but at times cumbersome to modify and update in a backward compatible manner.

III. Environment of Internet Streaming

At the time that DTV was designed the Internet was principally utilized for email. There was no real concept of Internet streaming media. The packet switched methodology of the Internet is structurally robust, but subject to variable delays and time variable bandwidth. In the roughly twenty five years since the development of DTV the availability of wideband Internet connectivity has expanded to encompass most of the world. The quality of service is variable, however the level of connectivity is impressive.

As wideband connections have become more widely available the use of the Internet for streaming media has become increasingly popular. In a largely parallel development to DTV the HyperText Markup Language (HTML) interface [10] and the browser became the primary means to interface with the various types of media available from the web. An extensive ecosystem enabling per user ad targeting and usage tracking grew up around this ubiquitous browser interface. This ecosystem is predicated on the HTML browser and depends on HyperText Transfer Protocol (HTTP) [11] for the delivery of the objects (files) populating the web page.

Early on a variety of streaming media methods existed in the Internet. Some of these were based on RTP [12] and RTSP [13]. The structure and practice of RTP bearing some considerable similarity to MPEG-2 TS based delivery. The assumption of constant delay in MPEG-2 TS based delivery resulted in a need to buffer in RTP based delivery. This buffering and the variability of available bandwidth often resulting in a less than satisfactory user experience.

The variability of the delay and available bandwidth lead to the development of technologies known broadly as adaptive HTTP streaming of which MPEG DASH is a leading example. The use of HTTP for broadband delivery enables deploying video streaming services utilizing widely available HTTP servers and caches, such that the deployment became easy, economic, and scalable. In addition, the use of HTTP also enables the transit of most firewalls, because the vast majority of unicast IP services depend on HTTP.

The PCR method of MPEG-2 TS depends upon the fixed delay assumption. The widespread availability of Coordinated Universal Time (UTC) [14] within the Internet allows MPEG DASH to eliminate the dependence upon PCR delivered master clock(s). This provides multi-fold benefits with respect to synchronous playback of media from multiple sources concurrently.

The environment in which these broadband streaming media services must play is an HTML browser, which, as previously described, is the ubiquitous environment for interactive user services in the broadband Internet.

The development of adaptive HTTP streaming environment included native support for both Media Source Extensions (MSE) [15] and Encrypted Media Extensions (EME) [16]. These provide access to a broad selection of media types and support for Digital Rights Management (DRM) within the context of an HTML 5 [17] interface.

IV. Evolution of DTV Broadcast Physical Layers

The first generation of DTV physical layers utilized MPEG transport packet as a native feature of the physical layer. As technology advanced the maximum physical layer packet sizes grew substantially larger than one hundred eighty eight bytes, e.g., IP is generally allowed to be at least fifteen hundred bytes. Further the concept of the mobile television and mobile access became a baseline feature in the design of DTV physical layers. The mobility enabled structure of these advanced physical layer(s) drove change in a number of significant manners.

The general structure of the physical layer was revised such that media streams can be accessed individually at the physical layer. This in the interest of reduced power consumption and hence suitability for battery operation. This is in contrast to first generation DTV physical layers which were assumed to have wall socket power and typically delivered a single complete multiplex carrying all the services from a single RF allocation.

An increase in physical layer FEC code block size to two kilobytes or sixteen kilobytes has resulted in enhanced physical layer performance, but also resulted in both MPEG-2 TS and IP being carried in layer two encapsulation. The core relationship of the physical layer to MPEG transport no longer exists in advanced DTV physical layers such as DVB-T2 [18]. This structural change has eliminated the inherent inefficiency of carrying IP transport inside of MPEG-2 TS as practiced with first generation DTV systems. The selection of IP delivery with potential application of header compression no longer carries an immediate protocol overhead penalty, as experienced with first generation IP via DTV physical layers.

V. Convergence of DTV and Streaming IP

A. Use Cases for Converged Services

Broadcast Streaming: All components are delivered via broadcast, however the user experience may be enhanced via a browser based user interface that allows for interactive application(s) to run in the same environment as streaming media. There are opportunities for enhanced revenue from the placement of targeted ads, especially when the viewing device has access to a broadband connection for reporting.

Broadcast NRT services: These are cached services that may be comprised of essentially arbitrary combinations of text, media, and graphics. As above, there are opportunities for targeting of content. The previous use case is likely comprised of a combination of cached NRT and real time content.

Broadband Streaming: All service components are delivered via a broadband connection. Key benefits herein being the opportunity to provide more narrowly focused content in a more efficient manner than via broadcast. This in turn enabling more precise targeting. There are also opportunities to maximize capacity, i.e., utilize broadband delivery when it is the most efficient method.

Broadband NRT: Unless there is a very substantial diurnal temporal variation in the network loading that provides substantial idle bandwidth for example overnight or known content item demand cached broadband delivery is not notably superior to on demand service. High demand makes NRT delivery appealing. Making this effective depends on prior reservations for content delivery or so called predictive delivery. The threshold transition from unicast to broadcast is impacted by network architecture. For example in an LTE network, the required number of users of a common content item may need to be only be one user per cell to make broadcast delivery more efficient than unicast for UHF reception.

Hybrid Services: The previous use cases focus largely on either high penetration content, i.e., that which is best served by broadcast delivery or content with narrow enough interest to be best serviced via broadband delivery. There is a class of services for which a single service may be comprised of both. These hybrid service components are for example secondary or tertiary language audio and text and or other more narrowly targeted service components.

The obvious conclusion from this diverse collection of use cases is that the ability to selectively utilize broadcast, broadband, or both enables a content provider to maximize both efficiency and potential revenue. The best way to address this potential is with a unified IP centric delivery method that leverages all the tools developed to support the broadband Internet.

B. Unified Broadcast and Broadband Receiver Stack

The use cases above suggest the need for unified receiver stack that can handle the full diversity of potential service types and delivery methods. Such a unification has significant advantages. Figure 1 above depicts such a receiver stack. This unified stack is organized, so as to allow clean interface among the various layers and the required functionality across and among the various delivery methods.

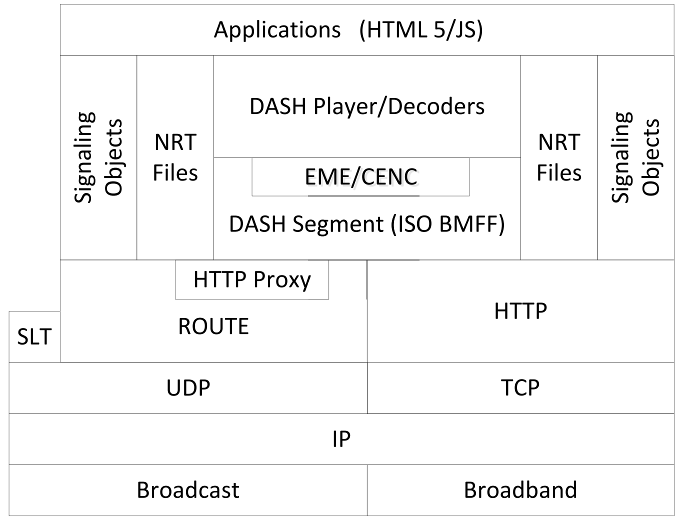

Fig. 1. Unified Broadcast, and Broadband Receiver Stack.

VI. ROUTE Object and Streaming Delivery Method

A. Motivation

In Figure 1 above the ROUTE delivery method is depicted in the broadcast delivery path. ROUTE is conceptually related to File Delivery over Unidirectional Transport (FLUTE) [19] in that it is a broadcast object (file) delivery method supporting the use of AL-FEC. A key distinction between ROUTE and FLUTE being that ROUTE is fully optimized for multiple object(s) and steaming media delivery, while FLUTE is more narrowly focused on single file delivery and was originally designed primarily for the NRT application. Key benefits of ROUTE being the ability to apply AL-FEC across multiple independent objects and support in the ROUTE receiver interface for faster channel change.

B. ROUTE Introduction

The ROUTE protocol provides generic application transport for any kind of object. It supports rich presentation including scene descriptions, media objects, and DRM-related information. ROUTE is particularly well suited to the delivery of real-time media content and offers many features including:

- individual delivery and access to different media components;
- support for layered media codecs by enabling the delivery of such an encoded service on different LCT transport sessions;
- support for flexible AL-FEC protection, including multistage FEC and object bundling;
- easy combination with MPEG DASH [2] enabling synergy between broadcast and broadband;
- fast access to media when joining a ROUTE session;

- highly extensible by focusing on IETF delivery mechanisms;
- enabling enhanced reuse of existing media format technologies, namely DASH and ISO Base Media File Format (ISO BMFF) [20];
- compatibility with well-established and deployed MPEG standards such as DASH and ISO BMFF formats;
- compatibility with existing IETF protocols and use of IETF-endorsed extension mechanisms.

C. ROUTE Feature Set

The ROUTE protocol is split in two major components:

- The source protocol for delivery of objects or flows/collection of objects.
- The repair protocol for flexibly protecting delivery objects or bundles of delivery objects that are delivered through the source protocol.

The source protocol is independent of the repair protocol, i.e., the source protocol may be deployed with or without the ROUTE repair protocol. Repair may be added only for certain deployment scenarios, for example only for mobile reception, only in certain geographical areas, or only for a certain service or type of service.

The source protocol is aligned with FLUTE as defined in RFC 6726 [19] as well as the extensions defined in 3GPP TS 26.346 [21], but also makes use of some principles of FCAST as defined in RFC 6968 [22], for example, that the object metadata and the object content may be sent together in a compound object.

In addition to the basic FLUTE protocol, certain optimizations and restrictions are added that enable optimized support for real-time delivery of media data; hence, the name of the protocol. Among others, the source ROUTE protocol enables or enhances the following functionalities:

- real-time delivery of object-based media data enabling fast channel change,
- flexible packetization, including enabling media-aware byte ranges, as well as transport aware packetization of delivery objects,
- independence of files and delivery objects, i.e., a delivery object may be a part of a file or may be a group of files.

The ROUTE repair protocol is AL-FEC based and enabled as an additional layer between the transport layer, e.g., UDP and the object delivery layer protocol. The AL-FEC reuses concepts of FEC Framework defined in RFC 6363 [23], but in contrast to the FEC Framework the ROUTE repair protocol does not protect packets, but instead it protects delivery objects as delivered in the source protocol, which in general provides better time diversity and is as a consequence more effective. However, as an extension to FLUTE, it supports the protection of multiple object(s) in one source block which is in alignment with the FEC Framework. Each FEC source block may consist of parts of a delivery object, as a single delivery object (similar to FLUTE) or multiple delivery objects that are bundled prior to FEC protection. ROUTE FEC makes use of AL-FEC schemes in a similar way to that defined in RFC 5052 [24], and uses the terminology of that document. The FEC scheme defines the FEC encoding and decoding, as well as the protocol fields and procedures used to identify packet payload data in the context of the FEC scheme.

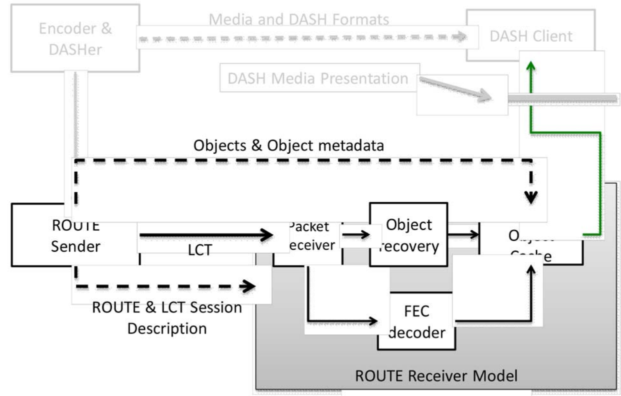

Fig. 2. Reference Receiver Architecture Model in ROUTE.

In ROUTE, all packets are LCT packets as defined in RFC 5651 [4]. Source and repair packets may be distinguished by:

- different ROUTE sessions, i.e., they are carried on different IP/UDP port combinations;
- different LCT transport sessions, i.e., they use different Transport Session Identifier (TSI) values in the LCT header;
- by the Protocol Specific Indication (PSI) field in LCT, if carried in the same LCT transport session. This mode of operation is mostly suitable for FLUTE-compatible deployments.

ROUTE defines the following features:

- source protocol including packet formats, sending behavior and receiving behavior
- repair protocol
- metadata for transport session establishment
- metadata for object flow delivery
- recommendations for MPEG DASH [2] configuration and mapping to ROUTE to enable rich and high-quality linear TV broadcast services. However, as an extension to FLUTE, it supports the protection of multiple object(s) in one source block which is in alignment with the FEC Framework as defined in RFC 6363 [23].

D. ROUTE System Architecture

The scope of the ROUTE protocol is the reliable transport of delivery objects and associated metadata using LCT packets. The objects are made available to the application through a Delivery Object Cache. The implementation of this cache is application dependent. This architecture is depicted in Figure 2, where in light grey is DASH, Black is ROUTE, and inside the grey box is ROUTE receiver model.

The normative aspects of the ROUTE protocol focus on the following aspects:

- the format of the LCT packets that carry the delivery objects,
- the reliable transport of the delivery object using a repair protocol based on FEC,

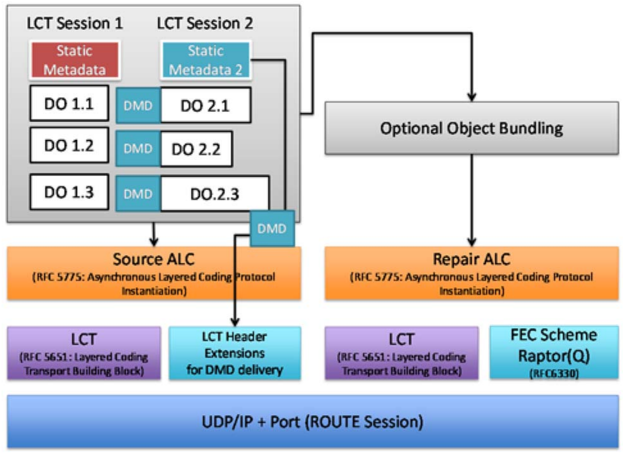

Fig. 3. Sender Operation of ROUTE Protocol.

- the definition and carriage of object metadata along with the delivery objects to enable the interface between the delivery object cache and the application,
- the ROUTE and LCT session description to establish the reception of objects along with their metadata, and
- the normative aspects (formats, semantics) of auxiliary information to be delivered along with the packets to optimize the performance for specific applications, e.g., real-time delivery.

One of the key issues addressed by using ROUTE is that the DASH media format as well as existing DASH clients may be reused. This architectural design enables converged broadband/broadcast services and enables established Over The Top (OTT) technologies (media codecs, metadata signaling, DRM, etc.) to be seamlessly adopted in broadcast services.

Figure 3 shows the basic sender concept. A ROUTE session is established that delivers LCT packets. These packets may carry source objects or FEC repair data. From a top down approach, a source protocol consists of one or more LCT sessions, each carrying associated objects along with their metadata. The metadata may be statically delivered in the Service-based Transport Session Instance Description (S-TSID) or may be dynamically delivered, either as a compound object in the Entity Mode or as LCT extension headers in packet headers. The packets are carried in Asynchronous Layered Coding (ALC) [25] using a specific FEC scheme that permits flexible fragmentation of the object at arbitrary byte boundaries. In addition, delivery objects may be FEC protected, either individually or in bundles. In either case, the bundled object is encoded and the repair packets are delivered separately. In combination with the source packets, this permits the recovery of delivery object bundles. Note that one or more repair flows may be generated, each with different characteristics, for example to supported different latency requirements, different protection requirements, etc.

DMD (Dynamic MetaData) is metadata that is used to generate Extended File Description Table (EFDT) equivalent descriptions dynamically at the client. It is carried in the entity-header in the Entity Mode and is carried in the LCT header in other modes of delivery.

The architecture supports different protection and delivery schemes of the source data. It also supports all existing use cases for NRT delivery, as it can be deployed in a backward-compatible mode to FLUTE.

E. Data Model

A ROUTE session is associated to an IP address/port combination. Each ROUTE session constitutes one or more LCT transport sessions. LCT transport sessions are a subset of a ROUTE session. For media delivery, an LCT transport session would typically carry a media component, for example a DASH Representation [2]. From the perspective of broadcast delivery of DASH formats, the ROUTE session can be considered as the multiplex of LCT transport sessions that carry constituent media components of one or more DASH Media Presentations [2]. Within each LCT transport session, one or more objects are carried, typically objects that are related, e.g., DASH Segments [2] associated to one Representation. Along with each object, metadata properties are delivered such that the objects can be used in application services which may include, but are not limited to, DASH Media Presentations, HTML-5 Presentations, or any other object-consuming application.

VII. Synchronization via Unified MPD

ATSC 3.0 utilizes a unified Media Presentation Description (MPD) which is described in 3GPP TS 26.346 [21]. This method allows delivery of service components via broadcast or broadband and delivery by both concurrently. The combination of broadcast and broadband defining a hybrid service.

The delivery of broadcast service components is available via either Media Delivery Event (MDE) [3] or Segment method. Both methods are relatively time inflexible, i.e., the media is delivery basically just in time and reconstructed according to a UTC synchronous Presentation timeline [2]. It is most straightforward and efficient to render the media as soon as this is allowed. The ROUTE receiver may hold off start of broadcast media playback briefly to assure that sufficient media has been received to allow start up with no subsequent stall. This process establishes a Presentation timeline in the DASH client and allows the DASH client to request the broadband components early enough to assure synchronous playback of service components from both delivery methods.

VIII. System Buffer Model for ROUTE/DASH

A. Event Definitions

A Data Delivery Event (DDE) occurs when a block FEC based MAC/phy delivers relevant contents of a specific physical layer block to a specific ROUTE session at specific time. Each DDE has a specific time slot at the physical layer and its reception time at the receiver is known by the physical layer scheduler.

A Media Delivery Event (MDE) data block is a collection of bytes that is meaningful to the upper layers of the stack for example the media player and codec(s). Figure 4 below

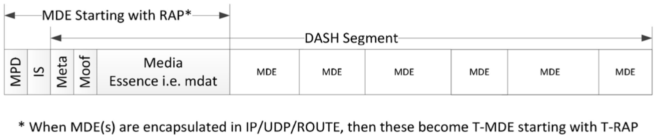

Fig. 4. Concept of MDE Starting with a RAP.

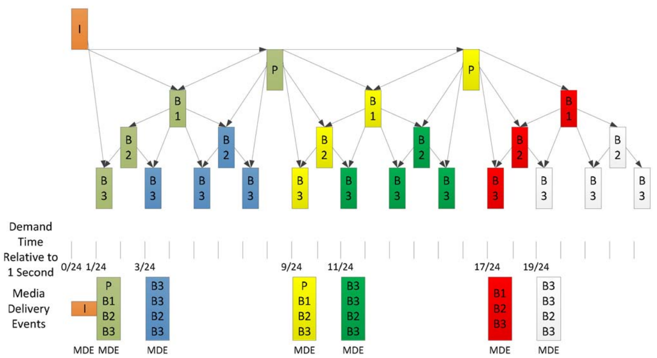

Fig. 5. Illustration of an MDE at the Video Level.

depicts the concept of an MDE starting with a Random Access Point (RAP), which becomes a Transport Media Delivery Event (T-MDE) starting with a Transport Random Access Point (T-RAP) once encapsulated in IP/UDP/ROUTE. In the Figure 4, "Meta" represents the collective set of meta-data boxes located at the start of a Media Segment, such as styp and sidx [2].

Figure 5 illustrates a "meaningful to upper layer" grouping of video frames. This example being an allowed structure for High Efficiency Video Coding (HEVC) [26]. In decode order, the orange I frame is playable by itself. Upon its receipt, the olive colored four frame group comprised of a B3, aB2, a B1, and a P frame become the next playable group. This pattern repeating for the remainder of the DASH Segment. A key aspect of an MDE data block is that it has a "required delivery time" to the DASH client at the interface to ROUTE. This does not need to be known at the receiver, but is required information on the infrastructure side for the scheduling of media on the physical layer.

B. ROUTE/DASH System Model Discussion

Much as MPEG systems [1] defines a buffer model for MPEG transport, there is a similar if somewhat simpler buffer model for ROUTE/DASH. Given the changes in the core methods of the physical layer as noted above there are certain differences in the corresponding ROUTE/DASH system buffer model. This model is depicted in Figure 6 below.

There are a number of distinctions for the ROUTE/DASH System model relative to the MPEG Systems model. The Data Delivery Events (DDEs) at specific t(i)s are not individual bytes, but rather the result of the delivery of a block of data by the phy/MAC at a specific time, i.e., a discrete data size in bytes at a discrete delivery time.

A TB_n buffer is for a specific ROUTE session, rather than an elementary stream. A ROUTE session may deliver multiple objects and related AL-FEC. The specific TB_n

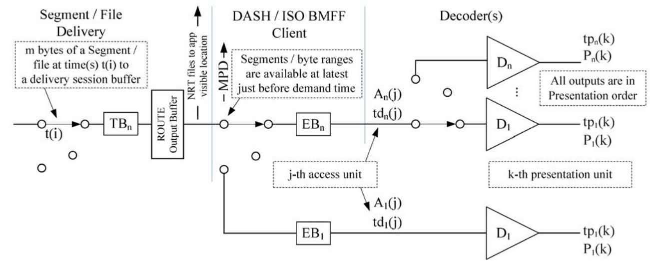

Fig. 6. System Model for ROUTE/DASH Delivery.

buffer applies to all the objects and AL-FEC as encapsulated by IP/UDP/ROUTE. The size of TB_n is derived via the attribute `@minBuffSize` as described in the ROUTE specification [3]. The size of TB_n is the sum of the `minBuffSize` values for each of the constituent LCT sessions in a ROUTE session.

All events in a ROUTE/DASH model are discrete time based, i.e., "m" bytes input to or "n" bytes output from the buffer at a specific time. Since all events are discrete time for ROUTE/DASH, there are no leakage rates specified.

The delivered objects for media services are very briefly buffered in the ROUTE Output Buffer before they are consumed by the DASH client. The minimum size of this buffer is slightly smaller than the corresponding TB_n, as the data in TB_n is wrapped in IP/UDP/ROUTE and may include AL-FEC packets. The output objects/files are of course both de-encapsulated and decoded.

The EB_n buffer is defined in MPEG Systems as an elementary stream buffer. This buffer, as applied to ROUTE/DASH, is associated with the ISO BMFF file handler, which holds data while it waits to be parsed to the decoders. Given that there may exist multiple object/file streams in an LCT session, there may be n total EB_n(s) associated with a given LCT session. There may also be "i" media types within a single ISO BMFF file stream and hence the multiple decoders D_n connected to each single instance of EB_n. The required size of EB_n is included as part of the definition of an appropriate DASH profile.

The task of scheduling media to the codec(s) in the receiver is exclusively the responsibility of the ISO BMFF file handler (within the DASH client). The scheduling of the objects/files to the ISO BMFF handler is part of DASH function. This described series of steps result in the ISO BMFF handler receiving Media Delivery Events, i.e., byte ranges or object(s) that make contextual and temporal sense and allows it to deliver samples (media frames) to the codecs, such that the Media Presentation timeline is met.

Unstated so far, but implicit, is that an LCT session may carry multiple content types, and that content types comprising a service may be available from a collection of separate LCT sessions, which belong to one or more ROUTE sessions. It is also significant that the ROUTE/DASH System model supports continuous streaming with live ad insertions, which may impose some specific constraints with respect to behaviors at DASH Period boundaries.

The operation of the ROUTE/DASH System is defined such that none of the constituent buffers TB_n, EB_n, or the

ROUTE Output Buffer area are allowed to overflow nor are the codec(s) allowed to stall due to lack of input media. Each buffer will start at zero and may likely go to zero briefly during operation.

t(i) indicates the time in UTC at which m bytes of the transport stream, i.e., a Data Delivery Event enters the target ROUTE transport buffer (exits top of the PHY/MAC layers.) The value t(0) is a specific time in UTC, which is known by the physical layer scheduler due to the systemic nature of the physical layer/MAC.

TB_n is the buffer size for the current LCT session(s) in the ROUTE session, including all transport of source object(s) and all related AL-FEC, as delivered in IP/UDP/ROUTE packets.

EB_n is the buffer in the DASH client that holds data between the reception from ROUTE transport output area, until the media is streamed according to ISO BMFF semantics to decoder(s) D_1-D_n.

$A_n(j)$ is the j-th access unit in a ROUTE delivery per codec. $A_n(j)$ is indexed in decoding order.

$td_n(j)$ is the decoding time in UTC of the j-th decoded media frame

D_n is the decoder for a specific media type per Service definition.

$P_n(k)$ is the k-th presentation unit for a specific content type. $P_n(k)$ results from decoding $A_n(j)$. $P_n(k)$ is indexed in presentation order.

$tp_n(k)$ is the presentation time in UTC of the k-th presentation unit

IX. Examples of ROUTE/DASH With AL-FEC

A. Introduction

There are a number of potential approaches to the use of AL-FEC delivery at the transport layer, which can be supported with ROUTE. Four use cases are described. Use of AL-FEC need not impact the channel change time for segment level playback. Choice of AL-FEC delivery configuration is another benefit of ROUTE flexibility. This section is not intended to be comprehensive with respect to applications of AL-FEC with ROUTE/DASH, just intended to provide a few examples.

B. AL-FEC for a Single Segment

Figure 7 illustrates three use cases for a case that repair data covers a single Segment.

In each of these examples the AL-FEC is computed across the entire delivery. The difference among the various examples is the scheduling of the FEC temporally relative to the source object(s) of the ROUTE delivery.

In case A, the Al-FEC is delivered before the Segment(s) of the ROUTE delivery. Playback with AL-FEC applied may start at the end of the ROUTE delivery which is coincident with the end of the Segment, as depicted.

In case B, the applicable AL-FEC is delivered immediately after the Segment(s). Playback with AL-FEC applied may start at the end of related object and FEC reception, i.e., the ROUTE delivery. This can be the lowest end to end latency for reception at segment level playback without AL-FEC applied.

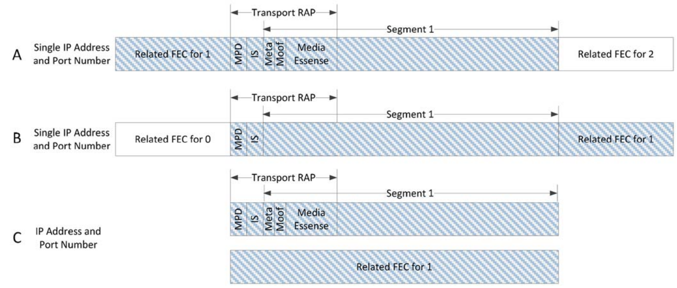

Fig. 7. Three Use Cases for AL-FEC for a Single Segment.

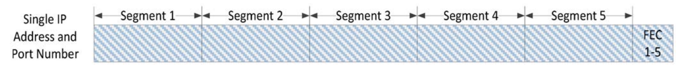

Fig. 8. Application of AL-FEC with Multiple Segments.

In case C, the applicable AL-FEC is transmitted concurrent with the applicable Segment(s) on the same Address and Port Number. The FEC has been interleaved with the source objects of the ROUTE delivery. Playback with AL-FEC applied may start at the end of related object and AL-FEC reception.

An appropriate start time for Segment/file level playback is known from the MPD as defined in MPEG DASH [2].

C. AL-FEC for Multiple Segments

Case D depicted below in Figure 8 is a fourth potential use case for AL-FEC with streaming media. In this example AL-FEC is operated across multiple media Segments in time. The application of AL-FEC across multiple Segments can achieve capacity gain.

All types of AL-FEC exhibit enhanced performance with increased time diversity for Rayleigh fading. Decorrelation time of Rayleigh fading is inversely proportional to RF carrier frequency. As a point of reference 3km/hr, which is well known Doppler rate for a pedestrian use case, is often a limiting use case for a physical layer. At 700 MHz, 3km/hr is a 2Hz Doppler with a corresponding 0.5 second decorrelation time.

An application of AL-FEC as shown below is potentially applicable for a DVR recording use case. AL-FEC as depicted below can be capacity achieving. It should be noted that any application of AL-FEC with less time diversity than the time diversity of the inner soft decision code at the physical layer will very likely result in loss of capacity.

A typical NRT use case may often be longer than five seconds in the time domain.

X. Signaling of Route/Dash Service Delivery

Service Signaling enables service discovery and provides descriptive information to allow the ATSC 3.0 receiver to acquire DASH-formatted service components over broadcast and/or broadband transport. It comprises two functional components: Bootstrap signaling via the Service List Table (SLT) and the Service Layer Signaling (SLS). These represent the

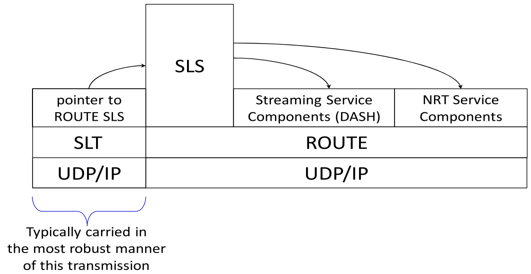

Fig. 9. SLT references to services via SLS.

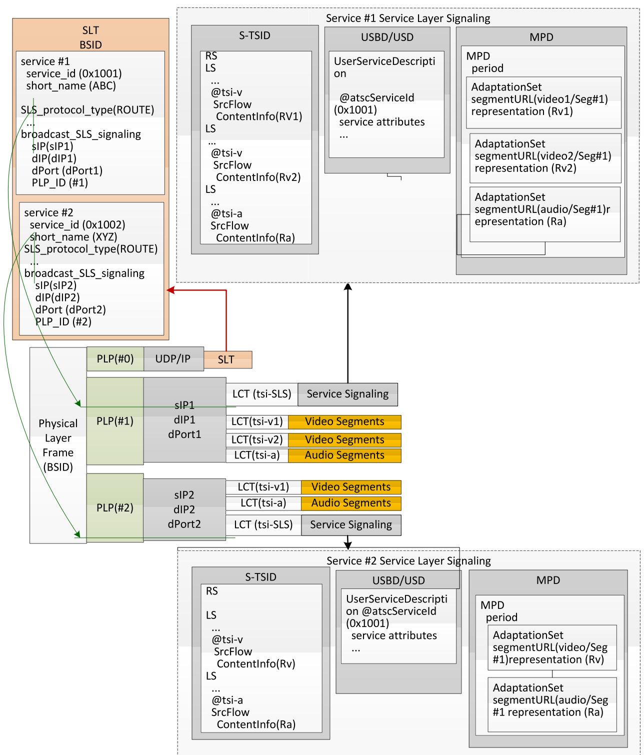

Fig. 10. Example Usage of Service Signaling to Support Bootstrapping and Service Discovery.

information necessary to discover and acquire user services. The SLT enables the receiver to build a basic service list, and bootstrap the discovery of the SLS for each ATSC 3.0 service. The SLS for each service describes characteristics of the service, such as a list of its content components and where to acquire them, and the receiver capabilities required to produce a meaningful presentation of the service. In the ROUTE/DASH system, the SLS includes the User Service Bundle Description (USBD), the S-TSID and the DASH Media Presentation Description (MPD). The USBD is based on the identically-named service description metadata fragment as defined in 3GPP-MBMS [21], with extensions that support ATSC 3.0 requirements.

A high-level architecture diagram showing the relationship between the SLT, the SLS, and the services described by the SLS, is shown in Figure 9.

An example use of service signaling for bootstrapping of service acquisition, and subsequent discovery of available DASH-formatted services delivered over ROUTE, is shown

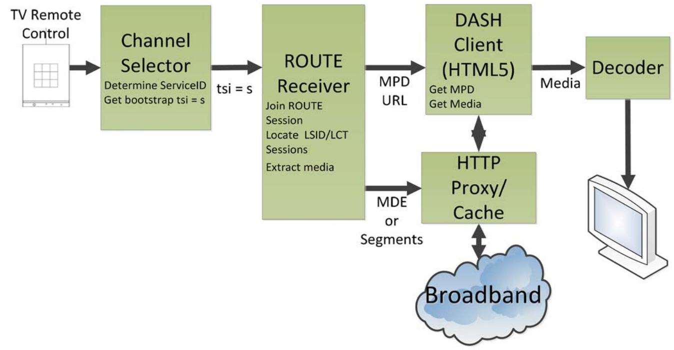

Fig. 11. ROUTE Receiver with HTTP Proxy/Cache.

in Figure 10. In this example, the SLT indicates the presence of two streaming services (services #1 and #2) transmitted over the RF broadcast channel. The SLT also provides the Physical Layer Pipe (PLP) identification and IP address/port information for acquiring the LCT sessions that carry service signaling data for those services. As shown in the example, service #1 comprises two broadcast video streams and an audio stream, each carried on a separate LCT session. Service #2 consists of a single video stream and an audio stream, carried on separate LCT sessions.

XI. Integration With Web-Based Applications

A key goal of ROUTE/DASH is to leverage existing web-based DASH applications (i.e., DASH clients) for both broadband and broadcast access. This is depicted as the common HTML5 layer in the protocol stack diagram of Figure 1. The web-based client is expected to leverage an MPD to fetch DASH segments, but it can also leverage the ROUTE receiver to determine the best form of access over which to retrieve those segments (e.g., cellular, WiFi, terrestrial broadcast, etc.). This is accomplished through the use of an HTTP proxy/cache resident at the ROUTE receiver. The manner in which this proxy/cache is integrated with the ROUTE receiver is depicted in Figure 11. The RF receiver selects the correct Physical Layer Pipe (PLP) by virtue of a well-known PLP ID to find service information. This service information disclosing the constituent PLPs and IP streams of the service. These IP streams containing ROUTE/LCT sessions. The content of the ROUTE/LCT sessions being DASH Media Segments for playback via the DASH client. Delivery of media components via broadband TCP/IP is also supported. In the case of hybrid services, both methods are used concurrently.

Conventional methods for designing the DASH client leverage HTTP GET commands for obtaining the MPD and DASH segments. For broadband connections, these requests can normally terminate either at the origin server or network edge caches (i.e., content delivery networks, or CDN's). However, in the case of ROUTE these requests are routed through the HTTP Proxy/Cache resident at the receiver. The proxy/cache can return the requested MPD or media segment directly to the client, independent of whether the requested data was retrieved from a broadcast or broadband connection. A popular approach to retrieving such data is the W3C standardized API XML-over-HTTPRequest [27] (XHR).

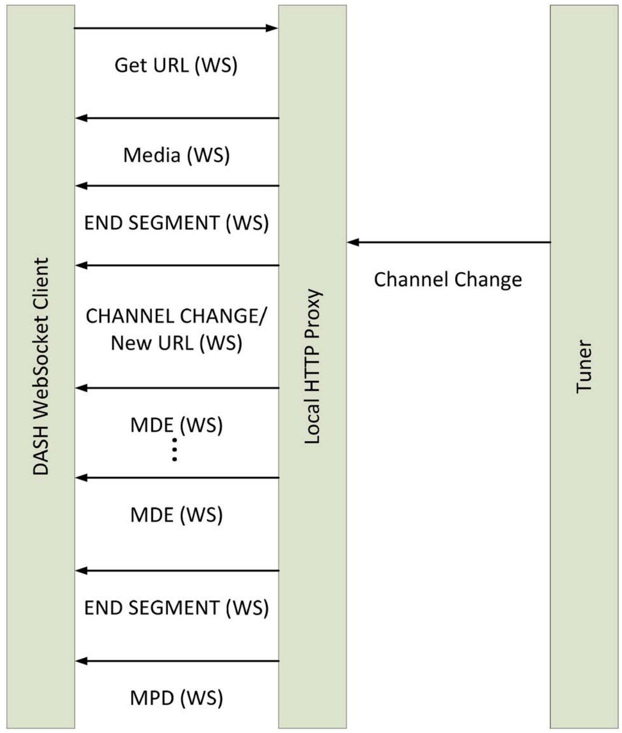

Fig. 12. Channel Change Message Sequence for WebSocket.

HTTP GET's have inherent latency due to the DASH client having to formulate the request for the segment. This involves parsing the MPD and issuing the XHR request. However upon a channel change event, this latency can affect the time from the user service selection (e.g., remote control selection of new channel) until the media from the new service is first rendered. Therefore, a push mechanism that is web-friendly might be used in place of HTTP GET's. WebSockets provide a potential solution.

WebSockets are a TCP-based sessions between a browser and a server, based on a client-server messaging model. Web sockets leverage traditional HTTP handshaking [28], but after completion of the handshake leverage a binary framing protocol. Like any other TCP session, the connection may be terminated by either endpoint. WebSockets allow a web application to maintain a persistent connection to a server, or in this case the local HTTP proxy/cache. More importantly, it offers an alternative to the traditional HTTP client-driven request/response model. Push messaging from the server is possible. Thus the proxy, being aware of channel change events, can immediately forward media through the proxy to the DASH client. The MPD is still useful to the DASH client, as the client can request additional media (referenced in the MPD) to augment the currently viewed programming.

A sample message flow is provided in Figure 12. In this example, the DASH client pipes its media requests through a WebSocket connection with the local proxy/cache. But the proxy/cache, upon being notified of a channel change (either directly from the tuner or indirectly through the ROUTE receiver) begins to push MDE's (Media Delivery Events, i.e., DASH subsegments) to the DASH client for immediate rendering. The new MPD URL is also pushed, but the example also depicts the new MPD being pushed in-band. This is accomplished through custom messaging that can be defined as a WebSocket subprotocol.

XII. Conclusion

As described ROUTE/DASH is a complete and appropriate solution for broadcast, broadband, and hybrid service delivery. The key aspects considered being straightforward integration of all methods delivery in a single IP centric platform, which enables clean layering, light weight, interactivity, extensibility, and content security. The overhead of ROUTE is essentially the same as the well-known and broadly deployed FLUTE, however the feature set of ROUTE is far richer, e.g., AL-FEC is provided across multiple objects and the integration with DASH, e.g., for rapid channel change is superior. These capabilities are provided on the widest possible range of IP centric device types and platforms, thereby maximizing the benefit to the broadcast community.

Acknowledgment

The authors would like to acknowledge the review and development efforts by Waqar Zia of Nomor, Gomer Thomas for LGE, Mark Eyer of Sony, Sejin Oh of LGE and demo development efforts led by Mr. Takahashi of Sony and Waqar Zia.

References

[1] *Information Technology—Generic Coding of Moving Pictures and Associated Audio—Part 1: Systems*, document ITU-T Rec. H.222.0|ISO/IEC 13818-1, ITU, Geneva, Switzerland, 2012.

[2] *Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*, document ISO/IEC 23009-1:2014, International Organization for Standardization (ISO), Geneva, Switzerland, May 2014.

[3] *Management and Protocols—Signaling, Delivery, Synchronization, and Error Protection*, ATSC Working Draft, document S33-1-384r18 ATSC 3.0, ATSC, Washington, DC, USA.

[4] *Layered Coding Transport (LCT) Building Block*, document IETF: RFC 5651, Internet Engineering Task Force (IETF), Reston, VA, USA, Oct. 2009. [Online]. Available: http://tools.ietf.org/html/rfc5651

[5] *User Datagram Protocol*, document IETF: RFC 768, STD 6, Internet Engineering Task Force (IETF), Reston, VA, USA, Aug. 1980. [Online]. Available: http://tools.ietf.org/html/rfc768

[6] *RaptorQ Forward Error Correction Scheme for Object Delivery*, document IETF: RFC 6330, Internet Engineering Task Force (IETF), Reston, VA, USA, Aug. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6330

[7] K. McCann. (Oct. 31, 2007). *Review of DTT HD Capacity Issues: An Independent Report From ZetaCast Ltd Commissioned by Ofcom Version 3.0*. [Online]. Available: http://stakeholders.ofcom.org.uk/binaries/consultations/dttfuture/annexes/report.pdf

[8] *ATSC Digital Television Standard, Parts 1–6*, document A/53, Advanced Television Systems Committee (ATSC), Washington, DC, USA, 2007. [Online]. Available: http://www.atsc.org/cms/standards/a53/a_53-Part-1-6-2007.pdf\

[9] *Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television*, ETSI Standard EN 300 744 V1.6.1, Jan. 2009.

[10] *Hypertext Markup Language—2.0*, document IETF: RFC 1866, Internet Engineering Task Force (IETF), Reston, VA, USA, Nov. 1995. [Online]. Available: http://www.ietf.org/rfc/rfc1866.txt

[11] *Hypertext Transfer Protocol—HTTP 1.1*, document IETF: RFC 2616, Internet Engineering Task Force (IETF), Reston, VA, USA, Jun. 1999. [Online]. Available: http://tools.ietf.org/html/rfc2616

[12] *RTP: A Transport Protocol for Real-Time Applications*, document IETF: RFC 1889, Internet Engineering Task Force (IETF), Reston, VA, USA, Jan. 1996. [Online]. Available: https://www.ietf.org/rfc/rfc1889.txt

[13] *Real Time Streaming Protocol*, document IETF: RFC 2326, Internet Engineering Task Force (IETF), Reston, VA, USA, Apr. 1998. [Online]. Available: http://tools.ietf.org/html/rfc2326

[14] W3C. *Date and Time Formats*. [Online]. Available: http://www.w3.org/TR/1998/NOTE-datetime-19980827

[15] W3C. *Media Source Extensions (MSE)*. [Online]. Available: http://w3c.github.io/media-source/

[16] W3C. *Encrypted Media Extensions (EME)*. [Online]. Available: https://w3c.github.io/encrypted-media/

[17] W3C. *HTML5 A Vocabulary and Associated APIs for HTML and XHTML*. [Online]. Available: http://www.w3.org/TR/2014/REC-html5-20141028/

[18] *Digital Video Broadcasting (DVB); Frame Structure Channel Coding and Modulation for a Second Generation Digital Terrestrial Television Broadcasting System (DVB-T2)*, ETSI Standard EN 302 755 V1.4.1, Jul. 2015.

[19] *FLUTE—File Delivery Over Unidirectional Transport*, document IETF: RFC 6726, Internet Engineering Task Force (IETF), Reston, VA, USA, Nov. 2012. [Online]. Available: http://tools.ietf.org/html/rfc6726

[20] *Information Technology—Coding of Audio-Visual Objects—Part 12: ISO Base Media File Format*, document ISO/IEC 14496-12, ISO/IEC, 2012.

[21] *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and Codecs (Release 12)*, 3GPP Standard TS 26.346 V12.2.0, 2014.

[22] *FCAST: Object Delivery for the Asynchronous Layered Coding (ALC) and NACK-Oriented Reliable Multicast (NORM) Protocols*, document IETF: RFC 6968, Internet Engineering Task Force (IETF), Reston, VA, USA, Jul. 2013. [Online]. Available: http://tools.ietf.org/html/rfc6968

[23] *Forward Error Correction (FEC) Framework*, document IETF: RFC 6363, Internet Engineering Task Force (IETF), Reston, VA, USA, Oct. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6363

[24] *Forward Error Correction (FEC) Building Block*, document IETF: RFC 5052, Internet Engineering Task Force (IETF), Reston, VA, USA, Aug. 2007. [Online]. Available: http://tools.ietf.org/html/rfc5052

[25] *Asynchronous Layered Coding (ALC) Protocol Instantiation*, document IETF: RFC 5775, Internet Engineering Task Force (IETF), Reston, VA, USA, Apr. 2010. [Online]. Available: http://tools.ietf.org/html/rfc5775

[26] *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, document ISO/IEC 23008-2:2013, International Organization for Standardization (ISO), Geneva, Switzerland, Nov. 2013.

[27] W3C. *XML HTTPRequest, Level 1*. [Online]. Available: http://www.w3.org/TR/XMLHttpRequest/

[28] I. Fette and A. Melnikov, *The WebSocket Protocol*, document IETF: RFC 6455, Internet Engineering Task Force (IETF), Reston, VA, USA, Dec. 2011.

Gordon Kent Walker received the B.E.E. and M.S.E.E. degrees in acoustical engineering from the Georgia Institute of Technology. He is a Vice President of Technology with Qualcomm Technologies Inc., San Diego, CA, where he provides technical leadership for the ATSC 3.0 project in corporate research and development. Previous technical leadership positions with Qualcomm have included LTE eMBMS, MediaFLO, and Digital Cinema projects.

Mr. Walker has been with General Instrument, M/A-COM Linkabit, Scientific Atlanta, and Harris Corporation(s) and is a member of SMPTE, AES, and IEEE. He has numerous granted and pending patents in the areas of digital communications, digital audio, and image compression and transmission. Products developed under his leadership include analog and digital video satellite receivers, digital microwave radios, audio signal processors, digital satellite modems, video scrambling, access control, and digital cinema equipment.

Thomas Stockhammer (M'98–SM'08) received the Dipl.-Ing. and Dr.-Ing. degrees from the Munich University of Technology, Munich, Germany. After acting as the founder and the CEO for ten years in a start-up, he joined Qualcomm in 2014 and holds a position as the Director of Technical Standards. He has co-authored over 70 patents with relevance to standards. He is the active and has leadership and rapporteur positions in 3GPP, MPEG, IETF, DVB, ATSC, and the DASH-Industry Forum in the area of multimedia communication, TV-distribution, content delivery protocols, and adaptive streaming. He has published over 150 conference and journal papers. He is a member of many different technical program committees.

Giridhar Mandyam received the B.S.E.E. degree (*Magna Cum Laude*) from Southern Methodist University (SMU) in 1989, the M.S.E.E. degree from the University of Southern California (USC) in 1993, and the Ph.D. degree in electrical engineering from the University of New Mexico (UNM) in 1996. He is the Head of Standards with Qualcomm Innovation Center (QuIC), a wholly owned subsidiary of Qualcomm Inc. He was a University Scholar and a Hyer Society Scholar with SMU. He was a Teaching Assistant of the Signal and Image Processing Institute with the USC. He studied under a NASA Fellowship with the UNM. From 1989 to 1994, he held positions with Rockwell International and Qualcomm Inc. He worked with Texas Instruments from 1996 to 1998. He was with Nokia from 1998 to 2006, where he held a number of research and management positions, including the Head of the Nokia Research Center cite in San Diego and the Chief Technologist for North America. Since rejoining Qualcomm in 2006, he was the Head of Research and Systems Engineering for Qualcomm Internet Systems before moving to QuIC in 2011. He is the inventor or co-inventor of over 35 issued patents. He has also published over 75 conference and journal papers, and five book chapters. He has co-authored the book entitled *Third-Generation CDMA Systems for Enhanced Data Services* (Academic Press, 2002). He was also the General Chair of the IEEE INFOCOM 2010 and the COMSWARE 2008, the TPC Chair of COMSWARE 2007, and the Industry Session Chair of WoWMoM 2007. He is on the Steering Committee of COMSNETS, a joint IEEE/ACM communications and networking conference. He has also served on numerous technical program committees. He has been an Editorial Board Member of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Ye-Kui Wang received the B.S. degree in industrial automation from the Beijing Institute of Technology, Beijing, China, in 1995, and the Ph.D. degree in electrical engineering from the Graduate School, University of Science and Technology of China, Beijing, China, in 2001. He was a Manager of Multimedia Standards with Huawei Technologies, Bridgewater, NJ, from 2008 to 2011. He held various positions, including a Principal Member of Research Staff with Nokia Corporation, Tampere, Finland, from 2003 to 2008, and a Senior Researcher with the Tampere International Center for Signal Processing, Tampere University of Technology, Tampere, from 2001 to 2003. He is currently the Director of Technical Standards with Qualcomm, San Diego, CA, USA. His research interests include video coding and multimedia transport and systems.

Dr. Wang has been a Contributor to various multimedia standards, including video codecs, file formats, RTP payload formats, and video application systems, developed in ITU-T VCEG, ISO/IEC MPEG, JVT, JCT-VC, 3GPP SA4, IETF, and AVS. He has co-authored over 500 standardization contributions, about 50 academic papers, and over 200 families of granted or pending patents. He has been an Editor for several standards, including HEVC, SHVC, HEVC file format, HEVC RTP payload format, SHVC/MV-HEVC file format, ITU-T H.271, scalable video coding file format, multiview video coding, IETF RFC 6184, IETF RFC 6190, and 3GPP TR 26.906.

Charles Lo received the B.S. and M.E. degrees in electrical engineering from Cornell University. He is a Principal Engineer with the Standards and Industry Organizations Department, Qualcomm Technologies Inc. He has been involved with various standardization activities, including 3GPP, 3GPP2, OMA, IEEE, and FLO Forum. He is currently an active participant in 3GPP on mobile broadcast services and in ATSC in the development of next generation broadcast TV standards, focusing on the application and service layer and IP transport areas. He has engaged in a variety of telecommunications research and developement and systems engineering roles, including fiber-optic transmission systems, video transmission via microwave subcarrier multiplexing, ATM SONET/SDH systems, and wireless multimedia communications.

Mr. Lo has worked with AT&T Bell Laboratories, Bell Communications Research, AirTouch Communications, and Vodafone.