A SCALABLE DISTRIBUTION SYSTEM FOR BROADCASTING OVER IP NETWORKS

R. J. Bradbury

BBC Research & Development, United Kingdom

ABSTRACT

The commoditisation of media consumption devices in recent years presents an opportunity to experiment with new content experiences that decompose traditional media content into discrete objects that can be recombined by receivers in many different ways. IP-based networks, both wired and wireless are expected to deliver these objects to consumers, but the need to scale them up to very large audience sizes, especially for simultaneous linear consumption, requires the development of advanced distribution technologies that are fit for this purpose.

INTRODUCTION

Time-shifted media consumption – especially the viewing of content on demand – continues to erode the viewing share of traditional linear television. However, the split in the UK is still around 15% (time-shifted) to 85% (linear) and while the trend towards on-demand consumption will probably continue for drama and entertainment, we believe that mass linear viewing will remain a mainstay for news, sport and other live coverage.

At the same time, valuable terrestrial spectrum is being reallocated from broadcasting to cellular radio telecommunications, meeting a perceived consumer demand for mobile data. Some of this capacity will be used for on-demand media consumption on the move. The scope for traditional broadcasters to introduce better-quality television services on today's digital terrestrial television platform is thus constrained. As broadcasters turn their attention increasingly to wired and cellular networks for the delivery of their linear services, the ubiquity of Internet Protocol technology presents an opportunity to converge distribution mechanisms around common networking standards.

As part of its public service remit to address all licence fee payers, the BBC has for a long time provided over-the-top Internet simulcasts of its linear television and radio channels, latterly as part of its successful iPlayer service. These are gradually migrating from proprietary technologies to standards-based formats like MPEG-DASH to ensure the widest possible reach, including commodity web browser clients and mobile devices as well as connected television sets and set-top boxes. Streams are encoded at several different bit rates and clients adapt dynamically between these as network conditions vary.

From an initial starting position where the BBC operated its own Internet streaming infrastructure, third-party Content Delivery Network (CDN) partners nowadays help us to meet an ever-increasing audience demand for IP-delivered media services. But the current unicast mode of delivery, where each viewer receives a unique stream from a CDN edge cache, is inefficient, costly and doesn't scale to audience sizes comparable with linear broadcast that we anticipate in the medium to long term.

Some fixed line telephony companies already distribute Pay TV channels over their managed networks using the more efficient multicast mode of transmission in which only one copy of a stream is transmitted over each network link. But current IPTV specifications are based on older MPEG-2 Transport Stream technology and do not allow for dynamic bit rate hopping.

BBC R&D is engaged in a research project attempting to combine the multicast transmission mode with the concept of dynamic stream adaptation. We have developed prototype senders and receivers, and have specified the network protocols necessary to demonstrate a robust end-to-end IP television transmission system. One of our design objectives is to target commodity receivers, and this includes not only traditional embedded software implementations, but also those that use generic web browsers for media playback.

Looking to the future, we have designed the system so that it can convey media “objects” of any kind with the highest resilience and the lowest possible network overheads.

Working with international standards bodies to adapt and reuse existing specifications where possible, we have created a means to distribute linear media objects at scale.

EXISTING STANDARDS

IPTV standards

The widely adopted DVB-IPTV specification [1] is aimed at conveying an MPEG-2 Single Programme Transport Stream across an IP network from a head-end source to many receivers. It achieves this scalability goal primarily by employing the technique of IP multicast packet replication, a feature commonly implemented in modern IP routers.

An MPEG-2 Transport Stream is designed to be carried over a quasi-error-free communications channel and decoded using a clock that is precisely synchronised with a broadcast source. IP networks, on the other hand, are designed with different reliability criteria. They are packet-switched and take advantage of asynchronous time division multiplexing to extract transmission efficiencies from the underlying network links. This results in links being oversubscribed and packets suffering from delay, jitter and even loss when a router’s packet buffer overflows. It is assumed that a higher-level transport protocol will provide packet loss detection and recovery where a particular application requires those features.

In the case of IP multicast, the only transport protocol widely available today is the User Datagram Protocol (UDP) which offers only rudimentary error detection, and no loss detection or recovery mechanisms. To overcome some of these shortcomings DVB-IPTV specifies the use of the Real-time Transport Protocol (RTP) [2] which provides some of the missing features, such as packet sequencing, and a timestamp field that enables clock recovery by the receiver. It additionally specifies the use of an audio-visual profile for RTP [3] originally devised for teleconferencing, and a payload format [4] for carrying up to seven MPEG-2 Transport Stream packets in an IP datagram on a typical network. To overcome the fundamental unreliability of IP networks, DVB-IPTV also specifies a Forward Error Correction (FEC) mode and a highly scalable retransmission service that allows individual receivers to request missing RTP packets.

Internet streaming standards

Over-the-top media streaming in the Internet was historically based on proprietary standards. Typically, streams were made available in unicast form and employ a variety of different media transport and control protocols, including RTP [2] and RTSP, but the audio and video CODECs carried were often vendor-specific. In the case of live streams, either the head-end system needed to be able to replicate the stream for each consumer and needed enough network bandwidth to serve those duplicate streams, or else specialist streaming media proxies had to be installed in the distribution network to achieve the desired scaling. Neither was an attractive option for ubiquitous access. Another factor limiting the success of these first-generation technologies was the limited availability of player clients for operating systems beyond the mainstream.

Making a stream available at a single bit rate can result in a poor end user experience. The unreliability of IP networks, especially packet loss resulting from traffic congestion, can cause playback interruption following an underrun event (so-called “rebuffering”) while the client attempts to refill its playback buffer. Clients typically insure against this risk by delaying playback to give them more leeway, but at the cost of latency compared with live.

A new idea to emerge in the past ten years is to make streams available at a variety of different encoded bit rates and to allow the end client to select the one best suited to the prevailing network conditions. This dynamic adaptation by the client requires corresponding switching points in the different encoded streams to ensure a smooth playback experience. Fluctuations in network performance are most acutely felt in cellular wireless data networks as the mobile terminal equipment moves from one cell to another, and it is no coincidence that the development of dynamic adaptive streaming technology has paralleled the widespread uptake of smartphones.

A number of proprietary technologies exploit this idea and combine it with the widely-implemented HTTP download protocol. Apple’s HTTP Live Streaming (HLS) is one of the most widely deployed examples and originally used segmented MPEG-2 Transport Streams to provide the required switching points. Slightly different approaches using fragmented MP4 files (more formally known as ISO Base Media File Format) are used by Microsoft for its Smooth Streaming format and by Adobe in its HTTP Dynamic Streaming (HDS) format. These companies, along with 3GPP, contributed their respective technologies to a standard for dynamic adaptive streaming called MPEG-DASH, published in early 2012 [5]. Recent efforts by MPEG to specify a Common Media Application Format (CMAF) based on fragmented MP4 offer hope at last of a unified packaging format for media, but failure to agree on one Common Encryption mode continues to hinder universal playback compatibility for protected content.

The trend for all these second-generation packaging formats has been towards the use of standard audio–visual CODECs such as H.264/AVC and AAC, both of which can be decoded in hardware on modern mobile devices. It remains to be seen whether H.265/HEVC will enjoy the same ubiquity in the future, or whether community-based efforts and royalty-free proprietary CODECs like VP9 stunt its uptake. The proliferation of different CODECs has implications on the number of different formats that content providers will need to publish in order to achieve universal coverage.

Problems with poor TCP throughput over long distances (resulting from a high bandwidth–delay product) have been addressed by serving content from edge caches placed strategically closer to the end user. Because the network round-trip time is reduced, TCP is able to achieve better throughput. Edge caching is made practical by the convergence of all parties on a standard download protocol: HTTP. Third-party Content Delivery Networks (CDNs) typically install caches in the core network of an Internet Service Provider to achieve the desired scale, and then charge content providers for the volume of content distributed. Some large content providers, including the BBC, have even installed their own caches in larger Internet Service Provider networks [6] [7] in order to have more direct control over both end user experience and distribution cost.

Multicast streaming standards

In cases where the same stream needs to be delivered to many receivers simultaneously, the multicast transmission mode offers efficiency savings over CDNs because the packet replication can be performed as a low-level function in a network router rather than by an additional high-level application function.

The Internet Engineering Task Force (IETF) has created a toolkit of protocol specifications for delivering media streams reliably over multicast IP networks. Recognising that different applications have different requirements, the Reliable Multicast Transport working group first specified a set of abstract protocol building blocks that could be combined in different ways [8] and then set about creating concrete protocol instantiations of these building blocks:

- **Asynchronous Layered Coding (ALC)** [9] is a protocol instantiation that provides reliability and massive scalability through the use of Forward Error Correction. It builds primarily on the Layered Coding Transport (LCT) building block.
- **File Delivery over Unidirectional Transport (FLUTE)** [10] uses the ALC protocol to deliver a file carousel over multicast. It introduces the higher-level concept of a File Delivery Session and a carousel manifest called the File Delivery Table.
- **NACK-oriented Reliable Multicast (NORM)** [11] is a different protocol instantiation that uses signalling of data loss by the multicast receiver (negative acknowledgements, or NACKs for short) to drive a bidirectional repair mechanism. Repair using Forward Error Correction is also supported as an option.

While technically strong, these protocols are complex and difficult to implement well. They are also quite different from the HTTP-based protocols in common use by web browsers embedded in consumer devices such as mobile telephones and tablet computers, a barrier to widespread adoption. They have, however, made some successful inroads in broadcast applications, as illustrated by the following three examples.

In the United States, CableLabs has developed and published an open specification for “multicast-assisted adaptive bit rate” over CATV networks [12]. In the CableLabs reference architecture, a centralised multicast controller is responsible for managing the complete end-to-end transmission chain from a content packager and a multicast server located at the cable head-end, to a multicast client embedded in the home gateway device. Segmented MPEG-2 Transport Streams or fragmented MP4 files are transmitted over multicast using a profile of the NORM protocol and any gaps resulting from missing packets are filled using byte range requests to an HTTP origin server, such as a CDN edge cache.

The 3GPP standards for cellular mobile telephony specify the use of FLUTE to deliver MPEG-DASH media segments over LTE networks as part of a Multimedia Broadcast/Multicast Service (MBMS) [13]. In the LTE Evolved Packet System architecture, a centralised Broadcast/Multicast Service Centre (BM-SC) is responsible for generating the FLUTE streams, and an MBMS Client in the mobile terminal equipment receives and processes them. The BM-SC explicitly controls which live streams are available via multicast and can signal a set of MBMS Clients to switch from a unicast operating mode to multicast when the number of consumers in a particular cell exceeds a configurable threshold (Multicast operation on Demand, or “MooD”). The use of an HTTP-based repair mechanism over a unicast bearer is also specified.

A variant of FLUTE called ROUTE (Real-Time Object Delivery over Unidirectional Transport) has been adopted by the Advanced Television Systems Committee (ATSC) in the United States as a proposed standard for delivering digital terrestrial television services over IP multicast as part of its ATSC 3.0 specification [14]. ROUTE is available as an alternative to the MPEG Multimedia Multiplexing Transport Protocol. The media essence is packaged as MPEG-DASH media segments in both transport protocol options.

BBC RESEARCH & DEVELOPMENT’S PROTOTYPE SYSTEM

In designing an object-based broadcasting system we had six high-level goals:

- **Reliability.** We want the playback experience to be as flawless as possible. We achieve reliability by combining a multicast mode of operation with a unicast fallback. FEC may not be needed, although this is the subject of future work.
- **Scalability.** We need to address a very large audience. We selected a mixture of IP multicast and conventional unicast CDN distribution as our delivery modes.
- **Reachability.** The ability to consume services on a wide variety of different terminal devices, including web browsers as well as embedded receiver devices such as smartphones, tablet computers, connected TV sets and set-top boxes.
- **Simplicity.** Multicast streams should be easily discoverable. The technology solution mustn’t be so complex that it cannot easily be implemented in commonly deployed embedded hardware.
- **Seamlessness.** The mode of reception (multicast or unicast) should be transparent to the application layer and the transition between these modes should be imperceptible to the end user.
- **Commonality.** We want to use identical encoding, packaging and distribution formats across unicast and multicast delivery modes.

We do not regard IP multicast as the primary mode of distribution in this system. The unicast mode, assisted by conventional CDN technology, remains the primary mode with IP multicast deployed selectively where the audience demand justifies its use. For example, a system operator may opt to distribute only the one or two most frequently requested video Representations (different encoded bit rates) of a channel via multicast based on analysing the cost of doing so relative to CDN distribution costs.

To that end, every object we distribute is uniquely addressed by a Uniform Resource Locator (URL) that is common across the two distribution modes. For an MPEG-DASH linear stream, individual media segments are numbered sequentially across all Representations in a particular Adaptation Set so that the “current” segment to be played back can be calculated by a client based on wallclock time. But this needn’t be the case for non-linear objects.

Multicast transport selection

In selecting a multicast transport protocol our goal was to map the request–response semantics of HTTP into a unidirectional mode so that a client can reference objects by their URL without needing to worry about whether the objects are being acquired from a multicast stream or requested from a server using conventional unicast retrieval. The main technical requirement in achieving this goal is to associate HTTP object metadata with the objects being transmitted in a multicast stream.

Our initial efforts centred on adapting RTP [2] to carry HTTP metadata. As with some of the other multicast technologies described earlier in this paper, this “retrofitting” proved somewhat cumbersome, in our case using RTP extension headers to convey the information required for successful reassembly and repair of transport objects. We next turned our attention to a new transport protocol – QUIC – developed originally by Google to speed up web access, and currently undergoing standardisation by the IETF [15].

QUIC (Quick UDP Internet Connections) is designed as a reliable transport protocol to rival (and perhaps eventually to replace) TCP. It is designed from the ground up to incorporate much of the accumulated thinking in transport protocol design gained from years of experimentation with TCP, and to correct some of its shortcomings. QUIC is a simple, reliable connection-oriented protocol built on top of UDP. Unlike TCP, where the pace of evolution is hampered by kernel development cycles, this allows rapid prototyping and testing of new features in user space without requiring modification of the operating system network stack, followed by a phased roll-out by operating system vendors. This, in turn, accelerates deployment and promotes uptake.

Crucial to our Use Case, from the very start Google specified a native mapping from HTTP request–response semantics directly onto the QUIC framing layer in a very natural and elegant manner [16]. A binary framing syntax inspired by HTTP/2 permits multiple logical request–response “streams” to be multiplexed into a single transport-level association without the “head-of-line” (front-of-queue) blocking problems that bedevil HTTP over TCP. The mapping also provides a handy “server push” mode of operation. This, combined with its UDP base, made QUIC an attractive choice for adaptation by us to multicast usage.

Description of system operation

Figure 1 illustrates our proposed system for large-scale broadcasting over the Internet.

A Head-end transmission function consumes resources from an HTTP origin server and, with optional transport-level encryption, segments them into packet-sized transmission units using the QUIC framing and packetisation layers. HTTP metadata, including the URL, is multiplexed in alongside each transmission object using standardised name–value pairs called “header fields”. Additional data integrity checks may be included in the QUIC packet layer at this point. Provision has also been made in the QUIC specification for the future addition of FEC codes. Finally, the QUIC packet stream is encapsulated in UDP/IP datagrams (IPv4 or IPv6) for transmission to a configurable multicast IP group address.

At the receiving end, the opposite process occurs in a function we refer to as a Client Proxy because it behaves as a caching HTTP forward proxy. (This is just one possible operating mode: others are equally valid.) The Client Proxy is responsible for subscribing to the appropriate multicast groups (using IGMP or MLD). QUIC packets are extracted from the payload of incoming multicast UDP/IP datagrams. They are integrity checked and decrypted (if transport-level encryption was used) before the QUIC frames contained in them are reassembled into transmission objects (the original HTTP resources).

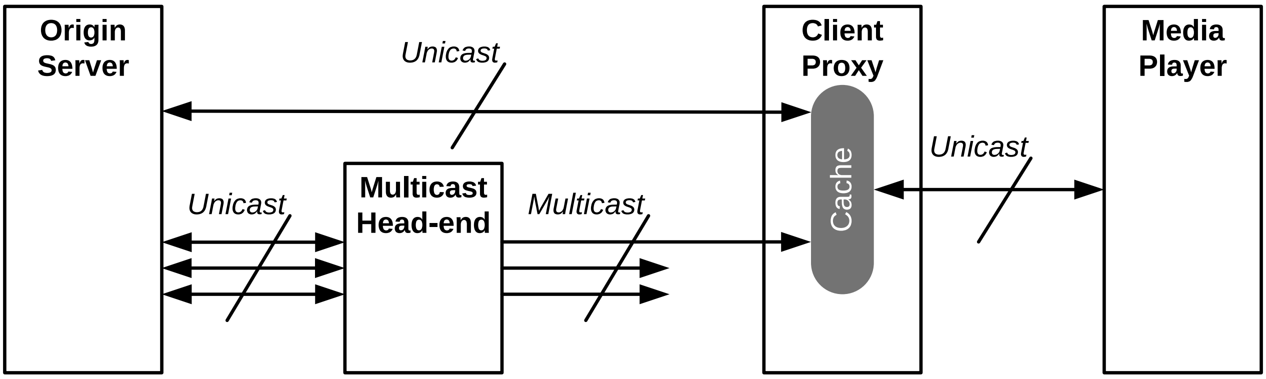

Figure 1 – High-level end-to-end system architecture

In our prototype implementation, we layer an end-to-end MPEG-DASH application on top of this basic arrangement. The Head-end system acts as a modified DASH client. First, it fetches and parses a DASH Media Presentation Description (MPD) corresponding to a particular linear channel. Then it retrieves media segments (the objects to be transmitted) simultaneously from one or more of the Representations described in the Presentation. The stream of media segments corresponding to each such Representation is serialised and transmitted to a different multicast IP group. The mapping between Representations and multicast groups could be configured in the Head-end software by some central system, or else conveyed in an extended MPD. In our prototype we instead convey the per-Representation configuration in the metadata of media Segments retrieved from the HTTP origin server.

A media player initiates a linear DASH streaming session by requesting the same MPD through its local Client Proxy. For each media Adaptation Set in the Presentation, the Client Proxy “thins” the MPD so that there is only one Representation. All responsibility for selecting a particular Representation is thereby ceded by the media player to the Client

Proxy. The Client Proxy also modifies the timing in the MPD to introduce a small delay before returning the MPD to the media player client.

Our implementation of the Client Proxy incorporates a simple HTTP cache backed by a modest amount of dynamic memory or non-volatile storage. The media player’s request for the first media segment of a particular Adaptation Set is routed through the Client Proxy. Since the requested segment is not available from its local cache (a cache “miss”), the Client Proxy simply forwards the request on to the HTTP origin server, and returns the response to the media player. It also caches the media segment for the benefit of any other clients consuming the same Presentation.

In parallel, the Client Proxy subscribes (if there is one) to the multicast stream corresponding to the currently selected Representation for that Adaptation Set and waits for the start of the next transmission object. As multicast packets arrive at the Client Proxy they are directed to a temporary buffer where they are reassembled into the original transmission object. If any multicast packets are lost in transit (for example, as a result of network congestion), and cannot be recovered by autonomous means (such as FEC) the reassembler is able to request the missing portions of the original transmission object via a conventional unicast HTTP “byte range” request to the origin server. (This could be achieved using HTTP/1.1, HTTP/2 or even HTTP over conventional unicast QUIC.) The reassembler uses standard HTTP metadata carried in the multicast transport protocol to identify the correct URL and the missing range of bytes. For efficiency, several byte ranges (disjoint or contiguous) can be packed into a single HTTP unicast “patch” request.

The Client Proxy places reassembled transmission objects into its cache. Because the timing of the Presentation was artificially delayed slightly when the MPD was returned to the media player application, there is an excellent chance that a given media segment will already be cached by the Client Proxy when the media player requests it (a cache “hit”). Unicast requests are thus limited to the start of a linear streaming session (for example, immediately following a user-initiated channel change event) and for “patching up” damaged transmission objects as part of a simple HTTP-based stream repair mechanism.

OPERATIONAL CONSIDERATIONS

A typical MPEG-DASH linear channel might comprise the following:

- An Adaptation Set offering a choice of several video Representations encoded with different CODEC profiles/levels, spatial resolutions, frame rates and/or bit rates.
- Adaptation Sets for different soundtracks (e.g. multilingual tracks, surround sound mix, object-based audio), each one perhaps offering a choice of several Representations encoded with different audio parameters at different bit rates.
- Adaptation Sets offering EBU Timed Text streams for live subtitles.
- Additional event streams providing user-facing metadata (e.g. information about the current and next programmes on the channel) or machine-readable metadata (e.g. to drive a companion screen application).

The operator of such a channel may choose to provide one, several or all of the encoded media Representations as multicast streams. The decision on which to offer via this mode will typically be based on an analysis of the operational costs measured against the benefits. This largely boils down to the number of consumers of a particular Representation. For each channel, there will be a crossover point where it is more cost-effective to provide the Representation via multicast than via conventional unicast. Conversely, below a certain threshold, the additional cost of conveying a multicast stream through the distribution network cannot be justified. For example, a channel operator may only choose to multicast the HD video stream and the main stereo soundtrack with all other Representations distributed via unicast only.

The decision on which Representations to distribute via multicast can be varied dynamically according to client demand. By monitoring unicast origin requests (probably CDN edge cache logs) in real time, a channel operator can detect when the threshold for enabling multicast has been reached for a particular Representation. In the prototype system described above, a multicast stream can be turned on or off at any time simply by manipulating HTTP headers at the origin server; in vertical deployments this could be a centralised control function of the network operator.

DEPLOYMENT SCENARIOS

The basic systems described can be deployed in a number of different configurations. In the simplest deployment architecture the Client Proxy is integrated into the receiver device alongside the media player and this tight coupling simplifies the operation of the two functions working in tandem. This is a very efficient deployment model because Layer 3 multicast packet replication by network routers is employed all the way to the end host. It does, however, require the device to have a multicast reception capability. While this may be the case for a mobile phone handset operating over an LTE cellular network, or a television set connected to a home router via an Ethernet patch cable, access to multicast packet streams over Wi-Fi networks is more troublesome because of limitations in commonly installed Wi-Fi access points [17]. Solutions to these technical problems exist [18] [19] but are still not yet widely deployed.

In an alternative deployment architecture the Client Proxy function is co-located with the Wi-Fi access point, typically in a home router device. The conversion from multicast to unicast is thus performed before interaction with the home Wi-Fi network. While it is less efficient to do the fan-out at Layer 7, the Client Proxy’s cache entries can now be shared by multiple media players watching the same linear channel simultaneously. From the perspective of a network operator, managing a single Client Proxy in each subscriber’s premises greatly simplifies a vertically-integrated deployment, especially if the operator supplies the home router device with the Client Proxy pre-integrated.

A third possible model for deployment is for a network operator to offer the Client Proxy function within its network, typically alongside the router nearest to the access network edge. In the context of recent innovations in edge computing, this third deployment model can be interpreted as an acceleration technique that uses Layer 3 multicast packet replication to feed a set of edge caches, thereby eliminating some of the latency that would otherwise be required to fetch media segments from an upstream mid-tier cache using a conventional Layer 7 unicast HTTP request/response interaction. Crucially, however, this is only an efficiency saving if the Client Proxy can be placed cost-effectively downstream of a “bottleneck” in the operator’s network. This may not be the case in some of today’s networks, but the current trend is for edge routers to be pushed deeper into both fixed and mobile networks, the latter as part of a more centralised Radio Access Network architecture [20].

STANDARDISATION WORK

BBC R&D's prototype multicast transport based on the QUIC packet and framing syntax has been published by the Internet Engineering Task Force as an Internet Draft [21]. We are continuing to work with IETF colleagues to promote the use of QUIC for multicast HTTP applications.

Meanwhile, the Digital Video Broadcasting (DVB) project has established a workstream looking at the standardisation of dynamic adaptive streaming over IP multicast. A number of companies came together in 2015 to draft commercial requirements and these have now been passed to the DVB's Technical Module for standardisation. BBC R&D is actively participating and hopes that some of its ideas in this area will make it into the final DVB specification.

REFERENCES

[1] Digital Video Broadcasting project (DVB), *Digital Video Broadcasting (DVB); Transport of MPEG-2 TS Based DVB Services over IP Based Networks,* European Telecommunications Standards Institute, TS 102 034, 2016.

[2] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550, July 2003. https://tools.ietf.org/html/rfc3550.

[3] H. Schulzrinne and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 3551, July 2003. https://tools.ietf.org/html/rfc3551.

[4] D. Hoffman, G. Fernando, V. Goyal and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video," RFC 2250, January 1998. https://tools.ietf.org/html/rfc2250.

[5] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), *Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats,* ISO/IEC 23009-1, 2012.

[6] F. Benedetto, "BIDI: The BBC Internet Distribution Infrastructure explained," 22 September 2016. http://www.bbc.co.uk/blogs/internet/entries/8c6c2414-df7a-4ad7-bd2e-dbe481da3633.

[7] Netflix Incorporated, "ISP Partnership Options," https://openconnect.netflix.com/en/delivery-options/.

[8] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd and M. Luby, "Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer," RFC 3048, January 2001. https://tools.ietf.org/html/rfc3048.

[9] M. Luby and L. Vicisano, “Asynchronous Layered Coding (ALC) Protocol Instantiation,” RFC 5775, April 2010. https://tools.ietf.org/html/rfc5775.

[10] T. Paila, R. Walsh, M. Luby, V. Roca and R. Lehtonen, “FLUTE - File Delivery over Unidirectional Transport,” RFC 6726, November 2012. https://tools.ietf.org/html/rfc6726.

[11] B. Adamson, C. Bormann, M. Handley and J. Macker, “NACK-Oriented Reliable Multicast (NORM) Transport Protocol,” RFC 5740, November 2009. https://tools.ietf.org/html/rfc5740.

[12] W. Jennings, D. Torbet and I. Wheelock, *IP Multicast Server-Client Interface Specification,* CableLabs, 2016.

[13] 3rd Generation Partnership Project (3GPP), *Universal Mobile Telecommunications System (UMTS); LTE; MBMS Protocols and CODECs,* European Telecommunications Standards Institute TS 126 346, 2017.

[14] Advanced Television Systems Committee (ATSC), *ATSC Candidate Standard: Signaling, Delivery, Synchronization and error Protection (A/331),* Advanced Television Systems Committee (ATSC), 2017.

[15] J. Iyengar and M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport,” Internet Draft, June 2017. https://tools.ietf.org/html/draft-ietf-quic-transport.

[16] M. Bishop, “Hypertext Transfer Protocol (HTTP) over QUIC,” Internet Draft, June 2017. https://tools.ietf.org/html/draft-ietf-quic-http.

[17] C. Perkins, D. Stanley, W. Kumari and J. C. Zuniga, “Multicast Considerations over IEEE 802 Wireless Media,” Internet Draft, 2017. https://tools.ietf.org/html/draft-perkins-intarea-multicast-ieee802.

[18] Institute of Electrical and Electronics Engineers (IEEE), *Local and metropolitan area networks — Specific requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications — Amendment 2: MAC Enhancements for Robust Audio Video Streaming,* IEEE 802.11aa, 2012.

[19] Institute of Electrical and Electronics Engineers (IEEE), *Local and metropolitan area networks — Specific requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications — Amendment 8: IEEE 802.11 Wireless Network Management,* IEEE 802.11v, 2011.

[20] A. Checko, H. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger and L. Dittmann, “Cloud RAN for Mobile Networks—A Technology Overview,” *IEEE Communications Surveys & Tutorials,* vol. 17, no. 1, pp. 405–426, 2015.

[21] L. Pardue and R. Bradbury, “Hypertext Transfer Protocol (HTTP) over multicast QUIC,” Internet Draft, February 2017. https://tools.ietf.org/html/draft-pardue-quic-http-mcast.