



Anytime, Anywhere, Best Quality: Multiscreen Streaming Workflows for Broadcasters

Chris Knowlton
Wowza Media Systems, LLC
Evergreen, Colorado, USA

Copyright © 2013 – 2014 Wowza Media Systems, LLC. All rights reserved.

Wowza and related marks are trademarks or registered trademarks of Wowza Media Systems, LLC. Third-party product names and related marks are trademarks or registered trademarks of such third parties. Use of third-party product names and marks does not imply any affiliation with or endorsement by such third-parties.

Abstract - *Today's consumer has developed an appetite for high-quality TV-like experiences on every device, from connected TVs to portable devices – how will you reach them all? As viewers embrace multiscreen lifestyles, broadcasters are challenged to deliver content to a myriad of devices over multiple protocols using evolving streaming formats. This paper explains how to deliver media to any type of screen while taking the viewing experience one step further – TV anywhere, anytime, at the best possible quality. You will learn about ways to stream protected content from contribution feeds to almost any IP-enabled endpoint. This paper also provides a history and overview of existing streaming media distribution models, and assesses the future of streaming media through developments such as Dynamic Adaptive Streaming over HTTP (DASH). By the end of the paper, broadcasters will better understand how to integrate a simple, cost-effective, high-quality multiscreen streaming workflow into their business model.*

INTRODUCTION

Much as fair-quality MP3 files were “good enough” to open up popular consumption of music on computers and pocket-sized audio devices, so too did fair-quality YouTube videos provide “good enough” experiences such that the paradigm for consuming video has moved ever more towards IP-enabled devices.

Newspapers and record labels didn't adapt quickly enough to maintain their share of eyes and ears as new technologies emerged. The good news for broadcasters is that there is still time to embrace the new paradigm for “anywhere, anytime” video consumption, perhaps first as a defensive move, and then as an offensive play, by using a multiscreen streaming media workflow.

STREAMING MEDIA HISTORY

How did we get from the original postage-stamp-sized low-quality video to the beautiful 1080p streaming and advanced interactive features that we see today? A little history may provide an informative context for a new streaming deployment.

I. Traditional Media Delivery

Traditional media streaming was developed in the mid-1990s as separate and largely non-interoperable proprietary formats. The most widely-adopted examples came first from Progressive Networks (now RealNetworks) and then Microsoft, followed later by Macromedia (now Adobe). For streaming content on the Internet, each of these formats eventually replaced the one that preceded it as the market leader.

In general, live streaming used User Datagram Protocol (UDP)-based solutions, which either skipped missing packets of data with visible glitches or used proprietary reliable-UDP protocols to reduce packet loss and video stuttering. On-demand streaming often used the chattier but ever-reliable Transport Control Protocol (TCP), which would not drop packets.

Unfortunately, the Internet infrastructure was not broadly ready for video streaming. Most viewers experienced stuttering video and the dreaded “Buffering” message in their media players.

By the mid-2000s, several changes had taken place. First, mobile devices and networks were being built on the new 3G (3rd Generation mobile telecommunications technology) standards. These provided efficient video streaming using Real Time Streaming Protocol (RTSP) to a broad range of mobile phones. Second, as Web traffic continued to increase, streaming started moving from traditional streaming protocols to Hypertext Transfer Protocol (HTTP) and its always-open port 80, making it easier to ensure that each packet traversed every intermediate firewall and arrived at its destination.

At about the same time, a new trend emerged, referred to as HTTP “progressive download.” This is the ability to use a standard Web server to deliver a media file to a media player with the logic to enable playback after downloading just several seconds of content, rather than waiting for the whole file to download, and to continue playback even as the simple download progresses from the Web server. The Apple iPod and iTunes were early contributors to this trend with audio files, followed a few years later by YouTube with video files.

II. HTTP Adaptive Streaming

In 2007, a new type of video delivery appeared from Move Networks - HTTP adaptive streaming. Unlike previous technologies from RealNetworks and Microsoft that tried to adjust video bitrates to the end user's bandwidth with mixed success, the new adaptive streaming worked very well. Bitrates and resolutions adapted seamlessly every few seconds, meaning users could finally experience uninterrupted video with decent quality on their screens, with most not noticing the temporary quality changes as bandwidth fluctuated.

The use of a sequence of small HTTP “chunks” - effectively tiny sequential progressive downloads of a few seconds of video - meant that any Web server could be used to deliver video. Better yet, every HTTP caching server on the delivery path could be used to scale out video distribution, which is especially critical for live events. On the downside, there were many HTTP video files per asset, sometimes tens of thousands. These were both difficult to manage in storage and in content delivery networks.

Microsoft made some significant improvements to HTTP adaptive streaming in 2008 with their introduction of Smooth Streaming, including the use of a “fragmented MP4” format based on the International Organization for Standardization (ISO) base media file format. This format enabled contiguous media files on disk to get “chunked” as they left the server. In 2009, Apple released a less-refined version of adaptive streaming called HTTP Live Streaming (Apple HLS) using the MPEG-2 Transport Stream container. Adobe followed in late 2009 with HTTP Dynamic

Streaming (Adobe HDS), which is very similar to Microsoft Smooth Streaming.

The following figure shows how adaptive streaming might switch or “adapt” between streams with varying quality, encoded at bitrates of 400, 900, 1300, 2000, 2500, and 3000 kbps, during the first 20 seconds of playback.

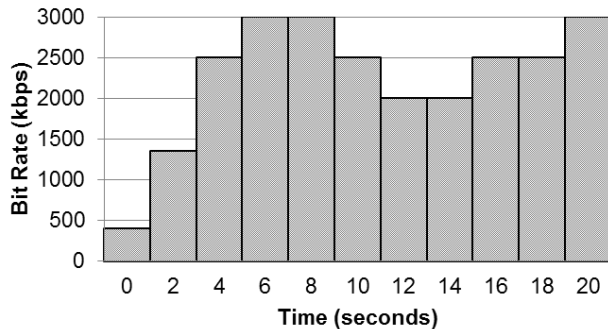


FIGURE 1- ADAPTIVE STREAMING EXAMPLE - BITRATE VS. TIME

Note that quality rapidly adjusts to the highest bitrate, and then moves downward for a few seconds to adjust to a slight drop in bandwidth before returning to the highest quality. Users may detect a slight reduction in detail crispness, but their viewing experience is not interrupted.

III. Codecs

With their traditional streaming formats, RealNetworks, Microsoft, and Adobe used video and audio codecs that were (or were perceived as) proprietary at a time when many streaming media customers and vendors were interested in moving to common codec standards. The most popular were H.264 video and AAC audio. Since 2007, almost all media streaming platforms have added support for these codecs, which are now used broadly for Blu-ray Discs, progressive download and streaming media delivery, and HD video broadcasting.

EXISTING STREAMING MEDIA DISTRIBUTION MODELS

I. Windows Media and Flash

For many years, broadcasters that adopted streaming media used Windows Media from Microsoft, and some still do today. Until recently, Windows Media provided a streaming platform that included an encoder to compress the source video, a server to distribute it, cross-platform media players to render it, and digital rights management (DRM) to apply encryption with associated business logic (e.g., expire after 24 hours, burn to DVD only one time, etc.).

As Microsoft reduced support for Windows Media playback on non-Windows operating systems in the mid-2000s, the cross-platform Adobe Flash Player became the de facto standard for watching unencrypted video in Web browsers. Adobe AIR was later introduced to extend Flash media delivery to desktop applications and mobile devices. Windows Media remained popular for DRM-protected

premium content and closed networks, including those run by cable operators and multiple system operators (MSOs), as well as telcos getting into the television services business.

II. RTSP/RTP Streaming

Many of the traditional streaming media platforms use the Real Time Streaming Protocol (RTSP) to control media delivery over the Real-time Transport Protocol (RTP). While RTSP streaming is used far less frequently now than HTTP for streaming from websites, it is still frequently used for certain use cases, including:

- Streaming video output from an IP camera
- Streaming media output from an encoder
- Streaming media to most mobile phones, including 3G, Android, Blackberry, and Symbian-based devices
- Streaming media to IPTV and HbbTV devices

III. MPEG-TS Streaming

MPEG-2 Transport Streams (MPEG-TS) were designed to deliver Standard Definition and High Definition television broadcasts over terrestrial, satellite, and cable networks. Some digital broadcast set-top boxes support incoming RTSP streams, but most set-top boxes supplied by service providers primarily support incoming MPEG-TS streams. Upstream, many linear broadcast encoders and video-on-demand (VOD) servers support delivery of MPEG-TS streams, but only a few broader-purpose streaming servers do.

IV. Smooth Streaming and HDS Streaming

With the rising popularity of commoditized HTTP adaptive streaming since 2009, some broadcasters have switched to either Microsoft Smooth Streaming or Adobe HTTP Dynamic Streaming to allow viewing of live/linear broadcasts and on-demand content, typically targeted at notebook and desktop computer screens. Each year has seen incremental improvements in these technologies that have improved the user experience. Examples of deployed production features include:

- **Dynamic streaming.** Content changes smoothly from one quality level (as defined by content bitrate and resolution) to another, typically every two seconds. Quality-level changes are driven primarily by information from the video player on the consumer device, which factors in the device's screen resolution, available bandwidth, and the frames-per-second rendering that it can sustain (taking into account the current CPU load and video processing capabilities of the device).
- **HTTP scalability.** HTTP adaptive streaming uses existing HTTP delivery infrastructures, including most public and many private content delivery networks (CDNs), so it can scale out very broadly. Traditional streaming typically relied on building out proprietary streaming server networks. Since streaming servers did not take advantage of relatively inexpensive core HTTP

infrastructures and network management tools, traditional streaming networks were often referred to as "overlay networks" and they were costly to implement. For most CDNs, the number of streaming servers was typically just a fraction of their HTTP servers, which often caused the streaming server network to overload during unexpectedly high peak traffic.

- **HTTP caching.** Adaptive streaming over HTTP can make use of HTTP caches to store popular content closer to consumers for short amounts of time, whether in the nearest CDN edge server from which the content came, in a cache within the user's Internet Service Provider, or in an HTTP cache set up at the Internet gateway of the business where they work to lower bandwidth costs.
- **Trickplay.** As with many playback experiences, adaptive streaming can support slow motion, fast-forward, and rewind on most playback devices.
- **Alternate video and audio tracks.** Includes dynamically switchable multiple-language audio tracks, multiple camera angles, director's commentary, etc.
- **Text streams.** These can deliver content such as subtitles, captions, chapter navigation, and script commands that can trigger delivery of advertisements, etc.
- **Rough cut editing.** Browser-based rough cut editing allows you to create content such as highlight clips for live events, either instantly or post-event. Viewers have higher engagement by interacting with new content during sporting events, etc. Being browser-based, no special editing stations are required, and the editing can be done from any location.
- **Ad insertion.** Many media players can insert pre-roll, interstitial (mid-roll), and post-roll advertisements. For linear content, this is typically created using the same adaptive streaming format and encoding profile as the premium content to ensure a consistent user playback experience.
- **Network DVR (nDVR).** Adaptive streaming can emulate the functionality of a digital video recorder by recording a linear stream at the streaming server. When a user "rewinds" into a linear stream, recent or popular content "chunks" can be retrieved from an upstream HTTP caching server or from the origin server. This functionality allows you to create an experience for "late joining" users, enabling them to either join the linear stream in process or start back at the beginning. An nDVR window can be set to specify how far back into linear content a user can rewind. A key advantage of nDVR is that there are no storage or hardware requirements on the user's device, whether it is a mobile phone or a set-top box. This allows you to deliver a DVR-like experience to any subscriber video screen with no new customer-premises equipment requirements.
- **Stream recording.** Most adaptive streaming servers allow you to record content as it flows through the

server for later on-demand playback. This feature can be available as part of the nDVR capability, as a separate "recording" or "archiving" capability, or both. Depending on the server, the recorded files can be delivered as on-demand streamed content from the same product with which they were created, or they can be delivered as progressive downloads from any Web server product.

- **Instant replay.** This nDVR-based feature provides a way for users to rewind some pre-configured duration, e.g., 15 seconds, into the content.
- **Digital Rights Management (DRM).** DRM permanently protects content and controls the way it can be used. A user typically signs into an account to be authenticated and authorized to retrieve a key to play back the protected content. Additional levels of security can be obtained by using key rotation, which can be set to periodically change the keys used within a given stream. Depending on the level of security required, you can selectively limit protection to particular frames, tracks, content sections, or streams.

V. HLS Streaming

Due to high demand for the ability to deliver video to iOS-based devices, most media server software providers have added support for packaging H.264/AAC content in the Apple HTTP Live Streaming format. Apple HLS does not yet have trickplay or some of the other aforementioned advanced features of either Smooth Streaming or HDS, but newer versions of the HLS specification continue to define additional capabilities. In the meantime, both Smooth Streaming and HDS are considered backward-compatible with HLS, as it is fairly easy to repackage Smooth Streaming and HDS content encoded with H.264/AAC into the simpler HLS format.

The greatest appeal of HLS is that it allows you to deliver content to iPhones, iPads, iPods, and Apple TV. In fact, to get video streaming apps accepted in the iTunes App Store for any of these devices, the apps must use HLS for streaming. Due to the popularity of iOS devices and the relative simplicity of HLS, the format is also being implemented elsewhere by third parties, such as on Android devices and set-top boxes, making it increasingly valuable for getting basic adaptive streaming content to a broader set of devices.

DEPLOYING EXISTING DISTRIBUTION MODELS - PARALLEL

1. Parallel Streaming Workflow

To reach the thousands of possible devices used by your customers requires supporting all of the different streaming formats described above. Until 2009, this required encoding content into separate formats and setting up parallel streaming workflows using products from multiple vendors. A simplified illustration of this can be seen in the following figure.

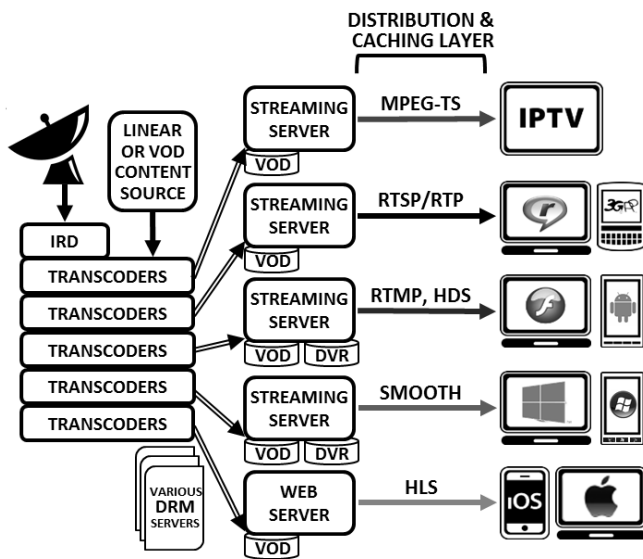


FIGURE 2 – PARALLEL STREAMING WORKFLOWS

II. Parallel Streaming Workflow Details

A parallel streaming workflow requires that you set up multiple encoders or transcoders, at least one per format. (For the purposes of this paper, assume that the incoming feed is already in a compressed digital format that requires transcoding.) In addition to converting one compressed audio or video format to another, the transcoders may also “transrate” the video for adaptive streaming. This process typically takes a single high-definition (“high bitrate”) version of the video and uses it to create multiple lower-definition (“low bitrate”) renditions with lower resolutions and higher compression ratios. Therefore, each transcoder would be delivering one or more linear streams to one or more appropriate streaming servers for that format.

If you are using adaptive streaming, you can provide nDVR capabilities for your linear streams. Similarly, you may simply archive any or all of the streams for later VOD use. In either case, each applicable streaming format will require storage for those new assets.

You would also typically have one or more transcoders per format doing batch-conversion of incoming on-demand assets for VOD delivery. For VOD assets, you might end up with dozens of files or renditions per title, representing multiple formats, resolutions, and quality levels needed to reach all endpoints.

The content protection system, whether conditional access, AES-128 network link encryption, or movie-studio-certified DRM, is typically different for each format. Multiple DRM license servers are often required to fully protect the different streaming formats and deliver keys to authorized consumers.

Even in 2009, many companies had not yet embraced adaptive streaming, which was still considered by many to be leading-edge. For those who had not taken that leap but wanted to stream, their workflows looked very similar to

Figure 2, simply replacing the adaptive streaming formats (i.e., Adobe HDS, Microsoft Smooth Streaming, Apple HLS) with their traditional streaming predecessors (i.e., Flash RTMP, Windows Media, and QuickTime RTSP).

III. Parallel Streaming Workflow Challenges

There are multiple challenges in running parallel streaming workflows, such as:

- Management.** Each format is delivered by a different streaming media server from a different vendor, each with its own management paradigm, programming language, and/or operating system requirements. To automate and scale streaming operations, you would need to integrate each into your workflow management system, including system operations, middleware, subscriber authorization, billing, monitoring, analytics, and so on. If using all-traditional streaming across a distributed network, the management is even more complex, as each format that could not make use of standard HTTP caching requires either fully independent management of an additional distribution or overlay network of (typically proprietary) streaming media servers, or a large investment in building a custom management layer across multiple proprietary streaming server platforms.
- Storage.** Although the core video and audio content is the same, and sometimes even uses the same codecs from one streaming format to another, each workflow requires that unique content versions be stored on disk, and where applicable, in the mid-tier and edge caching servers. As noted previously, you may end up with dozens of renditions of the same title in order to deliver the best possible video quality to all desired playback endpoints.
- Cost.** The cost of transcoders, servers, content protection systems, and storage for multiple formats is high. In addition, the burden of caching content that is essentially duplicative audio and video packets, but in different file packaging, throughout the distribution and caching layer either increases the costs or reduces the storage capacity for other content. If using all-traditional streaming across a distributed network, the costs are higher to add management of the overlay networks.
- User experience.** For parallel workflows, the consistency of the user experience varies between formats, depending on the available streaming bandwidth (especially for traditional streaming), the adaptive features enabled in each of the evolving formats, and the end user device playback capabilities.

DEPLOYING EXISTING DISTRIBUTION MODELS - UNIFIED

I. Unified Streaming Workflow

Starting in late 2009, streaming media server software providers began to expand their adaptive streaming capabilities by adding support for third-party HTTP adaptive streaming formats in their products. Although different

vendors support different formats, the common denominator among almost all vendors has been Apple HTTP Live Streaming. Although Apple HLS is not the most fully-featured of the adaptive streaming formats, it does allow each vendor to natively support playback on iOS devices, some Android devices, and some connected TV devices. For example, Adobe and Microsoft each support their own adaptive formats plus HLS, enabling them to reach most desktops and many devices. Several other server software and hardware companies, including Wowza Media Systems, have released products that support multiple traditional and adaptive streaming formats, allowing you to reach most video endpoints from a single server.

Multi-format server products have different ways of handling the workflow. When done well, the workflow for the formats is truly unified. At a fundamental level, this means that a) one core set of video and audio assets are created for each title, using a codec combination that works across adaptive streaming formats (i.e., H.264/AAC), and b) a single streaming server product then packages that core content as it leaves the server in slightly different “envelopes” to match the specifications for each adaptive streaming format.

The following figure shows a unified streaming workflow.

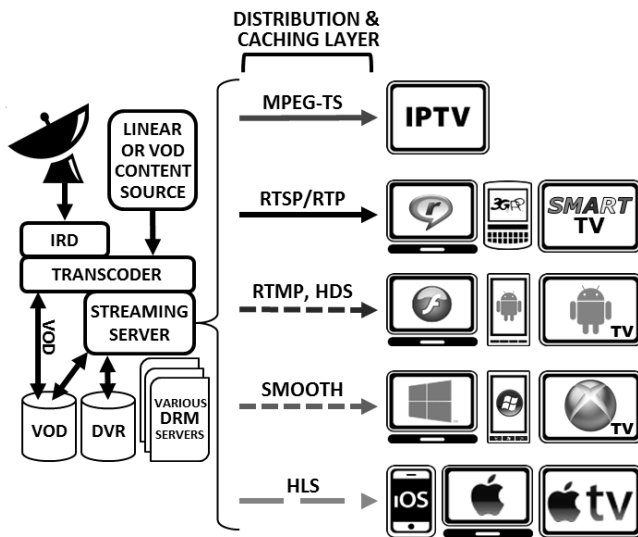


FIGURE 3 – UNIFIED STREAMING WORKFLOW

II. Unified Streaming Workflow Details

When ingesting from a live or linear source, the unified streaming workflow requires as few as one transcoder to re-encode the source content and create outputs at multiple quality levels for adaptive streaming to any endpoint.

For on-demand content libraries, there are three transcoding approaches to consider. The first is to do batch-transcoding in advance, which may include converting your

entire library into dozens of different quality-level renditions for each title, just as in the parallel streaming workflow. The second is to transcode content on-the-fly when it is requested, which can be ideal if you are delivering infrequently requested (“long-tail”) content. The third is to combine the first two approaches, in which you use batch-transcoding to pre-convert content that you know to be popular and set up on-the-fly transcoding to handle all the long-tail content.

You may be able to simplify transcoding even further by choosing a streaming server software product that also provides on-the-fly transcoding. This capability can be applied to both live and on-demand transcoding.

Whether live or on-demand, the content protection system is normally different for each format, as with parallel streaming. The one notable exception is that there are many examples in which Microsoft PlayReady DRM has been used to protect both Smooth Streaming and HLS content.

Some servers take this unified content strategy to higher levels of sophistication. For example, rather than recording and storing the content from a given live presentation in separate adaptive streaming formats, you may be able to store each quality level in its own normalized nDVR container for the duration of the nDVR playback window. On your origin server, which acts as the “point of truth” for all downstream edge and caching servers, only one full set of quality levels gets stored, rather than filling the origin with the same content in multiple formats. This nDVR content could then be saved as normal MP4 files for later on-demand playback.

III. Unified Streaming Workflow Advantages

The unified streaming workflow reduces complexity as compared to a parallel streaming workflow by simplifying streaming management. Using this architecture, you may need as few as one streaming media server product to deliver all of the streaming formats and protocols required to meet your customer and business needs.

Similarly, a unified workflow may reduce the number and types of transcoders used to just one product, if you standardize on a single video and audio codec combination such as H.264/AAC. As with having fewer server products, this provides additional cost and management benefits.

Finally, you may be able to even more fully realize infrastructure simplicity. Whereas a parallel streaming media workflow may have multiple transcoders and up to five server products, a unified workflow may allow you to combine the server and transcoder functions into a single streaming media server software product running on commodity or cloud hardware.

THE FUTURE OF STREAMING

I. MPEG DASH Overview

Just as customers and much of the streaming media ecosystem did not like having strong dependencies on vendor-centric streaming media solutions from companies

such as Microsoft and Adobe, neither do they like betting their businesses on vendor-centric adaptive formats such as Smooth Streaming, HDS, and HLS, over which they have no control. Although specifications have been published for all of these formats, none of them are industry standards. However, based on all three, a new international standard has been created: Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP (MPEG DASH). This new standard was ratified in November, 2011, largely due to the concerted technical input and cooperation from dozens of industry-leading broadcast, video, software, hardware, and services organizations that are members of the DASH Industry Forum (<http://dashif.org>). It was subsequently published as the International Organization for Standardization/ International Electrotechnical Commission (ISO/IEC) 23009-1 standard.

DASH draws from many of the best concepts in the vendor-centric adaptive streaming formats, including the use of segmented media files and a media presentation description file that describes key information necessary for a media player to request and play back audio and video segments from a server. As with Smooth Streaming and HDS, the ISO base media file format is a supported container, but DASH also supports the MPEG-2 Transport Stream container used by HLS. In addition, DASH enables a common encryption scheme for content protection and allows multi-language audio.

II. MPEG DASH Open Issues

Having an international standard will eliminate dependencies on vendor-centric formats and increase confidence in embracing adaptive streaming. However, it is still early for DASH, and there are a number of open issues to address. Two issues in particular are worth mentioning.

First, no codecs or encoding profiles have been specified, making it difficult to ensure interoperability across devices using the standard by itself. The good news is that the DASH Industry Forum has published “DASH264 Implementation Guidelines” that create an interoperability point for industry adoption. The guidelines define specific DASH profiles, codecs (i.e., H.264/AAC), and other key attributes needed to deliver both on-demand and live streaming services. This will eventually allow vendors across the ecosystem to offer fully interoperable products to broadcasters.

Second, although Apple contributed to DASH standardization, it has not yet indicated whether it will adopt DASH, or will instead continue refining HLS on its own, leading us down a bifurcated path of two de facto adaptive streaming standards. Fortunately, as with the ability to convert HDS and Smooth Streaming content to HLS today, it will likely be possible to convert DASH content to HLS tomorrow, so you will still be able to use an improved unified streaming workflow model based on MPEG DASH, as shown in the following figure.

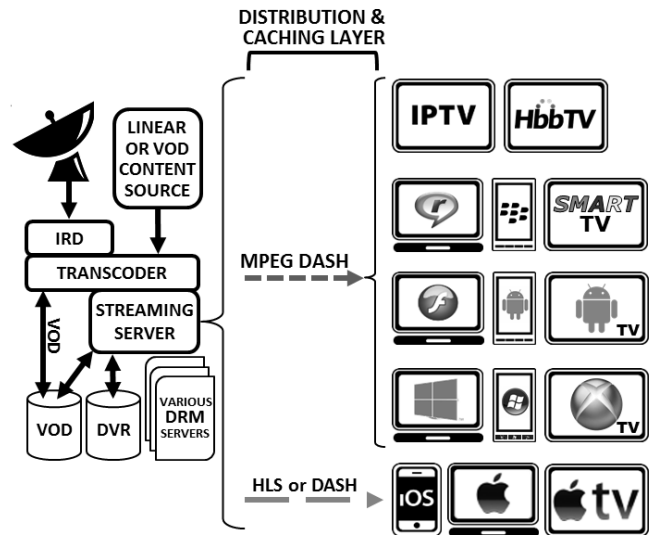


FIGURE 4 – MPEG DASH STREAMING WORKFLOW

III. MPEG DASH and HTML5

There has been much hype about video playback using HTML5, the next version of Hypertext Markup Language (HTML) that is a World Wide Web Consortium (W3C) standard and the core language of the Web. The promise is that a single new video element in the HTML code of a webpage or application will allow you to deliver video to any HTML5 browser or application, regardless of device or operating system. What you don't often hear about is that there are many video capabilities and features not supported yet by HTML5, including cross-platform adaptive streaming, live streaming, DRM, consistent playback and codec support across browsers, advanced video playback features, etc. In general, HTML5 allows you to deliver MP4 files using progressive downloads to most – but not all – Web browsers. "The State of HTML5 Video" (<http://www.longtailvideo.com/html5>) is a helpful and regularly updated report that tracks the state of HTML5.

The powerful combination of MPEG DASH and HTML5 would bring together the latest industry standards for both Web-based development and video delivery. Recognizing this, the W3C HTML Working Group developed Media Source Extensions to add the power of features such as adaptive streaming and nDVR to HTML 5.1. Expect to see proof-of-concept demonstrations at the National Association of Broadcasters (NAB) Show and other trade shows.

Note that finalization of the HTML 5.1 standard is not expected until late in 2016. With demand for interoperable HTML5 video streaming already high, chances are good that there will be widespread HTML 5.1 feature adoption in Web browsers, operating systems, applications, and devices before then, although full functionality and interoperability across them will likely not arrive until months or years after finalization. Until then, DASH streaming in non-HTML-5.1

DASH-capable media players will still help simplify streaming media infrastructure requirements.

IV. MPEG DASH and HbbTV

Hybrid Broadcast Broadband TV (HbbTV) is a standard for connected TVs and set-top boxes that seamlessly blends delivery of broadcast television with IP-based content delivered over an Internet connection. The goal is to allow users to enjoy rich experiences on their televisions using a single remote to access content such as traditional linear broadcasts, online video services, programming guides, interactive advertising, catch-up TV, video on demand, games, social networking, and more.

A critical aspect to the increasing adoption of HbbTV is that it is based on several existing standards, including the Open IPTV Forum (OIPF), Consumer Electronics Association (CEA)-2014 (CE-HTML), W3C (HTML 4/5), and the Digital Video Broadcasting (DVB) Application Signaling Specification (ETSI TS 102 809). HbbTV itself was approved by the European Telecommunications Standards Institute (ETSI) as ETSI TS 102 796 in June, 2010.

In November, 2012, a new version of the HbbTV standard (HbbTV 1.5) was approved. This update incorporates MPEG DASH as the designated format for delivering adaptive streaming content to Internet-connected TV devices.

HbbTV is sometimes promoted as a pan-European specification and is heavily adopted throughout Europe. However, HbbTV is also gaining acceptance outside of Europe, with deployments planned or underway in countries such as South Korea, Japan, China, and the United States. In

addition, numerous name-brand television and set-top-box manufacturers are producing HbbTV-enabled devices, providing both choices and healthy competition in markets where HbbTV has been rolled out.

As HbbTV adoption continues, taking MPEG DASH with it into more real-world deployments in Europe and beyond, it further bolsters the use of DASH as the standard of choice for streaming media.

CONCLUSION

Progressive download and traditional streaming did much to drive the adoption of video consumption over IP networks. As bandwidth, quality, and playback capabilities have improved, consumers have grown to expect a great video experience on all of their video-capable devices, from phones to connected TVs.

The current state-of-the art for any-screen delivery is the unified streaming workflow model, which greatly reduces complexity and cost over traditional streaming media delivery methods. This model is rapidly replacing older, traditional-streaming-only models, and greatly expands reach and quality to users on their devices of choice while simplifying infrastructure.

Going forward, DASH adoption appears to be poised to accelerate rapidly, especially with its incorporation as the streaming media format in other Web and connected TV standards. Expect the unified streaming workflow to be subsumed by the MPEG DASH workflow, which takes the best parts of unified streaming and further reduces deployment complexity while maintaining any-screen reach and increasing feature parity across those screens.