Reexamining The Design of Next Generation Video Streams

Yasser Syed, PhD
Comcast Distinguished Engineer
T&P/VIDEO Architecture Group
Comcast Cable

Abstract

The popularity of on-demand viewing means that content is being stored and retrieved more than ever before. Oftentimes, it is video playout on handheld devices, and resultant battery drain, that impacts performance more so than any lack of bandwidth. Similarly Content Delivery Networks/CDNs can impair viewing because too many copies of video packets are being stored. Yet for certain live events, the audience is larger than ever before.

Video streams traditionally have been designed for compression efficiency and with a specific focus on linear systems. This has affected codec selection, transmitted resolution, GOP length, random access, conditional access and hardware. Such linear-related restrictions have done a good job at balancing good linear video experience with good content delivery. But while traditional video streams may be suitable for linear systems, they may actually cause scaling issues for newer approaches that deliver IP video content over CDNs, mobile networks, mesh systems, and other unmanaged infrastructures.

This paper examines how video codec design could be optimized for the rising gamut of IP delivery mechanisms, and how a re-examined codec could impact the design of next generation video streams using IP delivery.

INTRODUCTION

The IP transformation of video systems is already happening with a transition between Linear-based QAM systems to CDN-based IP systems. Along with this transition is the shift to non-linear viewing models that can be viewed not just on television displays, but also on multiple, smaller display devices owned by the consumer. This results in increased demands for storage capacity -- which will, in the future, rival the bandwidth delivery constraints already occurring in IP unicast delivery.

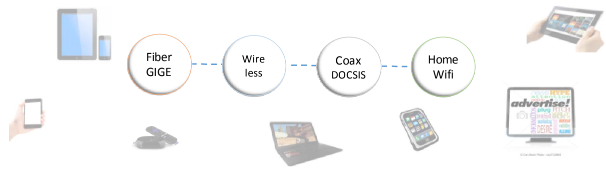

Figure 1- Delivery Over Hybrid Networks

Along with this IP transformation is the expansion of new features for video services. This includes features based on new display technologies, such as Ultra High Definition/ UHD and High Dynamic Range/ HDR. These additional features can increase the amount of content versions that are stored as well as adding pressure to bandwidth capacity.

In addition, video services are also transversing across hybrid delivery networks (cable, WiFi, mobile—both over managed and unmanaged bandwidth) and are expanding into different devices (television, PCs and portable devices such as laptops, tablets, and smart phones). The over-arching benefit of switching to an IP infrastructure is robustness in delivery design, despite variations in networks and device types. This allows for networks and devices to improve without significantly changing the delivery approach.

DESIGNING TRANSITIONS

To aid in this transition to an IP infrastructure, a separated transcoder/packager structure is being developed. The transcoder prepares the content into multiple Elementary Streams (ES) wrapped by a delivery transport structure (in this case, an MPEG2-Transport Stream/TS structure). The transcoder also marks up the stream to create virtual segments (or Encoder Boundary Points /EBPs) that can carry timing, boundary, and labeling points. The stream output from the transcoder is known as an Adaptive Transport Stream

(ATS). The ATS is then ingested by a packager, which can be co-located with the transcoder, or located further down in the network. The packager uses the marked-up points to create a manifest. It also segments the content into pieces in a single media format, similar to a DASH Manifest with MPEG-TS segments. This format is then called by the client or CDN and is transmuted in a lightweight process into the particular client-requested manifest type and media segment format (HLS, DASH, HDS, HSS, ISOBMFF).

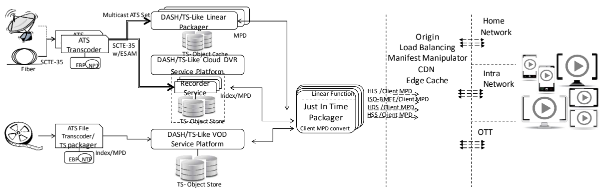

Figure 2- Transcoder & Packagers in the IP delivery Network

This transcoder/packager structure assists in the IP transformation in three ways: 1) by allowing access to the content in a stream format for point-to-point, intra-network distribution (push model); 2), by creating a segmented and indexed format for long term storage (pull model w/ manifest); and 3) by modification to whatever client-specific adaptive streaming format needed for loading into CDN (pull model w/ manifest). The combination of these formats allows us to use existing equipment to encode and check for video quality, store content in a single format, and only output in several formats when content is pulled out to the client delivery network. This also allows us a pathway to start utilizing a pull-model-based IP delivery architecture.

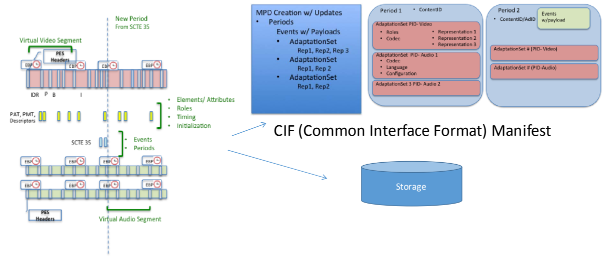

Figure 3- Conversion from Push to Pull Model Using Packagers

MANAGING OPTIMIZATIONS

It is usually the case that operational factors in switching to new technologies or different delivery systems can impede adoption of new delivery designs. While moving through the IP transformation, both for non-linear viewing and for new service expansion there will be a need to focus on optimizations in the following areas:

- Reducing the amount of identical copies and perceptually similar variations of content stored in the network (storage efficiency)
- Localizing content traffic demands through the use of well-designed CDN and edge servers (bandwidth efficiency)
- Reducing the volume of serially-dependent transactional traffic in the network (throughput efficiency)
- Extending battery lifetime of portable devices through the optimization of decoding complexity (display life optimization)
- Increasing the quality bitrate ratio of content streams to allow for video service expansion (future feature optimization)

In the new delivery system, the tradeoff between how to store content and how to transmit content will need to be considered simultaneously, because being efficient in one medium may cause waste in the other. Additionally, with IP delivery, the requests caused by content selection will increase, because each fragmented piece of content is requested separately -- and if missed, then re-requested -- which subsequently will cause more content to be sent across the network. With respect to devices, extending battery lifetime is not only important to consumers, but operationally important as well, to avoid disruption of services simply because the device powers down frequently. Lastly, with the additional features of 4K/8K and HDR that are being added to IP video services, better video compression will be needed. Better compression will help in avoiding any lack of perceived functionality of the service (i.e. UHD with Stereo Audio) because of its inability to scale up with increased traffic. Any of these factors can easily affect the deployment of video services over IP delivery systems, if the system does not accommodate for them, and as traffic volume increases.

USING SEGMENTED CONTENT

To address these transformations, Adaptive Bitrate (ABR) Streaming technologies are being used in combination with CDN delivery structures. The benefit of this is the ability to deliver bitsize chunks of video in a transactional delivery protocol for data IP packets (e.g. HTTP). This allows for the payload to be switched through a manifest indicating the fragment options that are available to the player. It converts video into packetized data IP delivery that can easily be routed through data handling devices using port 80 designations.

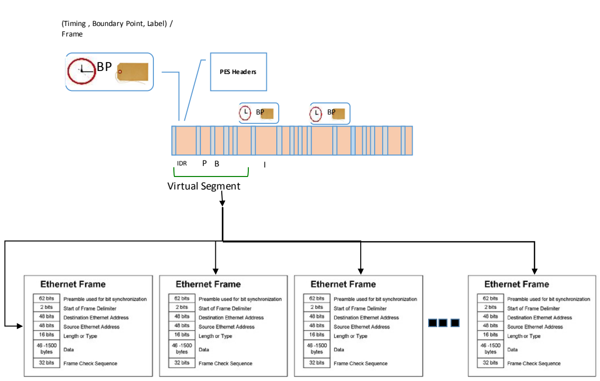

Figure 4- Fitting a Segment into IP Encapsulation

The video stream itself is encoded into a specific codec format, which in itself is a NAL (Network Adaptation Layer) unit. The packetized video stream is buffered and synchronized for both compression and constant framerate playback.

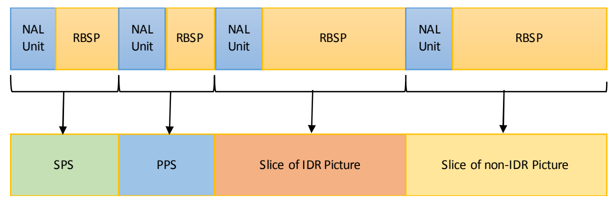

Figure 5- Video Stream Packetization

When an individual client makes a serial request for a video stream, in a best effort network, the trick to ensuring smooth playback is to deliver that stream --which has been encapsulated into multiple IP frames -- while being fast enough to keep the video CPB buffer filled. The rate of the buffer being filled is dependent on the fragment rate, since the whole fragment needs to be received in order to load it into the buffer. If the network is not fast enough to keep the buffer filled, then a lower bitrate fragment is requested, if the manifest knows that it is available to the player. The benefit of ABR is that it allows a consistent way to do video streaming over different network conditions and devices.

In first-generation ABR technologies, the design of the video streams was not drastically changed from the GOP design used in linear QAM streams. Rather, it was adapted and restricted to fit ABR technologies. At the beginning, in the mid-to-late 2000s, some of the earliest implementations, video Streams were adjusted around the following guidelines:

- GOP duration and the fragment duration were tied together to make the linear stream or VoD file easily divided into fragments by a packager

- All content segments were of the same duration
- All segments were independently decodeable
- All ABR streams had to have the same GOP structure (or a decimated version of it) to allow easy bitstream switching at every fragment
- All streams were constant bitrate streams to allow the fragment to be switched to a known bitrate
- All segments were meant to be delivered to a client player (so a server/ client relationship)

Improvements came later to allow for one or more GOPs to be contained within a segment duration. This was accomplished by creating a signaling construct that identified specific IDR frames in the stream, so as to indicate the beginning of the segment and allow the codec design flexibility. The result was increased quality. Notably, not all segments had to be the same duration (most of them, but not all of them), which allowed for splice-accurate ad insertion. Lastly, the delivery of segments did not have to be just from a server to a client player. This introduced the idea of storing content in one ABR format and allowing a packager to rewrap it into other ABR formats. Further work is occurring in second generation ABR technologies and corresponding video stream conditioning, particularly in standards development organizations and forums such as MPEG-DASH, MPEG Systems (virtual segmentation), JCT (HEVC), the DASH Industry Forum (DASH-IF), and SCTE WG-7.

REEXAMING VIDEO STREAM DESIGN

As this second generation ABR work is occurring, it is timely to consider how the video stream could be adapted to fragmented IP delivery, rather than vice versa. It comes down to how we handle video streams of the future. Granted, there are transitory issues to consider, but rethinking how we will use video streams in an IP delivery infrastructure could bring about a fresh approach in the design of the video stream. The rest of this paper asks questions that encourage the reader to relook at the design of the video stream, including what activities are already occurring in these areas, and how some modifications may help in handling new demands in next generation systems.

What if we accessed content differently?

With fragmented content, manifests, and the existence of several MBR (multi-bitrate) streams for content, there are many possibilities for new approaches to content access. Content could be indexed to handle program starts/end, chapter starts/end, and ad starts/ends. These can be indicated in the manifests through the period mechanism or url mechanisms, and triggered in the encoded stream through the SCTE 35 constructs and labeling af_descriptors (ATS/DVS 1196). The player could access content through a set of content indices, as described in the manifest. This allows the viewer to intelligently access the content by skipping to meaningful points (seek vs scan) -- but it is also dependent on how much of the content is indexed before and during the transcoding step.

At a more myopic level, random access of stream fragments takes on a new meaning with the existence of MBR streams. With several streams present, random access could now be defined differently in each separate MBR stream. If one needed quick access into the content, a representation could be selected where random access would happen at each fragment boundary. In another representation, the quality of the stream could be improved by reducing the number of random access points in the stream. Bitstream switching can still happen between representations but not at every fragmen. The effect of this could be twofold: 1) Either longer duration fragments, or fragments that may not be independently decodeable because they are dependent on the preceding fragment in sequence (playthrough on higher bitrate representations) or 2) shorter duration fragments that are independently decodeable (random access on lower bitrate represenations). This MBR approach could re-introduce useful "new/old" concepts, such as open GOP, which was once needed to increase the quality of the stream by allowing for larger fragments to be broken up, which subsequently allows some flexibility in fragment delay at the player buffer. The new requirement would determine some level of alignment of different streams in the MBR set.

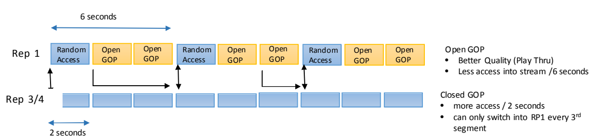

Figure 6- Setting different Random Access Points for Each Representation

Another consideration in video access is the level of handling of the stream. Because video services need to extend beyond just a server/player distribution, and into hybrid networks and backbone infrastructures, there are several intermediary devices that need to handle, store, or route the fragments or stream. These devices are not necessarily involved in the playout of the content. Partial encryption of the stream allows for this handling, where header or marker information is in the clear, but actual video frame content is encrypted. This "partial reveal" allows for stream handling to happen, without the access for playout. Common encryption takes this a step further, in that it doesn't need two different sample variations of the remaining parts of the still-encrypted content for storage and distribution. This allows for devices to decrypt and re-encrypt for each mode, while still allowing DRM policy to be applied in both client-based and transactional models. This reduces the number of different versions of content stored, while also reducing the amount of transactional traffic in the network to decrypt and re-encrypt content.

What if we stored differently than we packaged?

Traditional storage systems were mostly file-based designs, which kept the number of items in the storage hierarchy small -- even though each item was rather large and non-uniform in size. Access to files at faster speeds required increasingly complex read/write/ or caching strategies. Additionally, file integrity required redundant approaches, like RAID-striped storage, to preserve content in the event of storage equipment failure.

New object-based storage systems have started increasing in popularity, given recent technology improvements that allow handling of lots more items, while flattening the storage hierarchy, and increasing the speed of read/write access to content. File integrity in object-based storage technologies moved from RAID-based striping approaches to the storage of redundant copies, in different pieces of hardware, across federated-based storage systems (which is very cloud-friendly). Also, moving to an object-based system allows for faster read/write access, which can accommodate easier parallel retrieval of content.

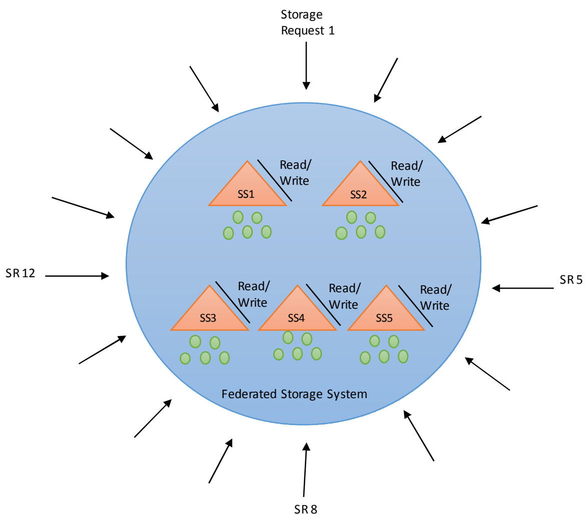

Figure 7- Access into Federated Storage Systems with multiple parallel requests

Object-based storage systems can be very accommodating when retrieving small content fragments in parallel request situations. In file-based solutions, a byte offset is needed to retrieve the content fragment if it is part of a bigger file. In object-based solutions, such retrieval could be simplified, made parseable, or altogether eliminated in certain designs. With storage design, and especially in object store designs, there is just as much awareness of segment size as segment duration. If duration-varying segments were used more, this could help in storage designs and segment retrieval. Methods to do this could include changing the placement of virtual segmentation boundaries, to accommodate more frames in a segment, or to assign different partitions to the streams. A drawback in using time-varying segments is that it would increase the size of the manifest.

What if we "switched" on something other than bitrate?

In traditional adaptive streaming, the client player monitors the CPB buffer and sends out a message to the server to switch either the representation or the profile to a higher or lower bit rate, based on current network conditions. The player tends to behave in a greedy fashion, always asking for more, unless it risks stopping the display of the video -- but at the same time, this response is not immediate. The issue here happens when thousands of players are all acting greedily, which can cause pulsating waves of bitrate demands. This is usually mitigated through the use of policy enforcement on the server side. Policy enforcement can make adaptive streaming systems more complex and very state-aware, so simplifying the strategy and making it less invasive could help to avoid what is a burdensome policy enforcement strategy.

How could players assist in this issue? Simply by acting less greedily. This could be done by making players more informed -- not just switching representation levels solely based on bitrate. For instance, if the quality of the representation level is known to the player, it could decide not to ask for the highest-level representation, especially if the quality is not discernably better. With this approach, the system could be saving bits that may be needed by other players. Other dimensions to this would involve a decoding complexity factor as well. Other approaches could address the design of representation levels by defining them in terms of constant quality factors. This, in turn, would bring in video coding concepts like Variable Bit Rate /VBR (alright capped VBR), which is proven to save on bits. Using a quality level to define stream representation could still allow the player to switch based on bitrate (lower quality usually means lower bit rate, or even lower complexity).

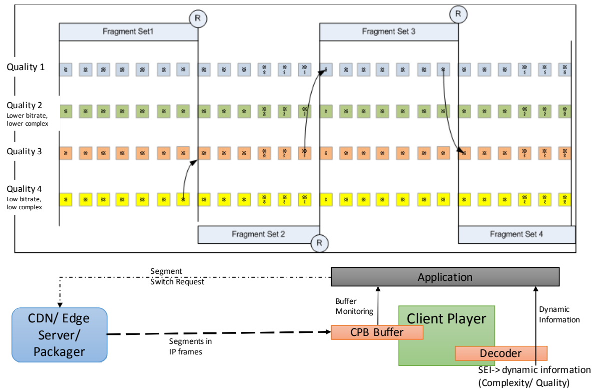

Figure 8- Switching Between Different Representation using more than Buffer Monitoring

Some of these approaches work under the assumption that information about switching choices resides only in the manifest -- but this doesn't have to be the only place. The stream itself could carry information that could be useful to the decoder. This information could be more dynamic than information carried in the manifest. If the player can access information that is available at the time of decode, the player could then make better decisions about switching to another representation. For instance, if a complexity indicator was sent as a Supplemental Enhancement Information / SEI message, the SEI message could be used by a player with limited battery power to switch to a lower quality representation, which would, in turn, extend the device's battery life. Another example could embed a dynamic quality indicator in the SEI that could be used by the player to rachet up or down a representation level, and manage its bit rate demands. Yet another example could indicate the brightness characteristics (nits) of a scene, which could again be used by the player to alter the display, so as to save on battery power. (Actually, the application of SEI messaging is a popular concept for both HDR display adaptiveness and "green" metadata for battery conservation.) These type of approaches can work well in reducing the "greediness" of players, while not demanding an intrusive policy enforcement strategy.

CONCLUSIONS

With next generation systems, there is an opportunity to re-examine stream design in light of the different demands being placed on the network for IP delivery. Specifically, we need to consider not just bandwidth and storage demands, but now display device battery lifetimes, and related delivery across a hybrid of networks. Optimization, and especially optimization of content handling, is key in reducing some of these demands, especially as the network scales up.

This paper provides some different perspectives in terms of processing, storage, and handling of content at a video stream level -- especially in a system with ATS streams, packagers, and adaptive streaming technologies. Other efforts exist to reduce the complexity of storage, including the standardization of media segment formats though MPEG's CMAF (Common Media Application Format) efforts, but these improvements can be orthogonal without issue. In addition, we need to consider traffic optimization and latency, which can be worked through strategies relating to CDN architectures, multicasting, edge servers, and data caching. Ultimately, such an approach could yield a transition plan and delivery system for on-demand content over IP that is both robust and stable enough to sustain across many types of networks -- from cable to WiFi to mobile, mesh, and others, as they materialize.

REFERENCES

[1] ISO/IEC 13818-1:2013, Information Technology - Generic coding of moving pictures and associated audio information - Part 1: Systems.

[2] ANSI/DVS 1196, Adaptive Transport Stream, www.scte.org

[3] ANSI/SCTE 214-1, MPEG DASH for IP-Based Cable Services, Part 1: MPD Constraints and Extensions, www.scte.org

[4] ANSI/SCTE 214-2, MPEG DASH for IP-Based Cable Services, Part 2: DASH/TS Profile, www.scte.org

[5] ANSI/SCTE 214-3, MPEG DASH for IP-Based Cable Services, Part 3: DASH/FF Profile, www.scte.org

[6] ITU-T Recommendation H.264 | ISO/IEC 14496-10:2012, Information Technology - Coding of audio visual objects - Part 10: Advanced Video Coding.

[7] ITU-T Rec. H.265 | ISO/IEC 23008-2:2014 – MPEG-H Part 2: High Efficiency Video Coding

[8] Y.F. Syed and A. Giladi, Hybrid Content Generation Workflows: from MPEG-2 TS to Adaptive Streaming and ISO-BMFF, proc of INTX 2015

[9] ISO/IEC 23009-1:2014 Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats.

[10] DASH-IF Interoperability Points, Version 3.0, http://dashif.org/wp-content/uploads/2015/04/DASH-IF-IOP-v3.0.pdf

[11] F. Fernandes, X. Ducloux, and S. Cheng, "Proposed White Paper on Green Metadata" M37090 MPEG 113 Meeting, Geneva Switzerland, Oct. 2015.

[12] ISO/IEC 23001-7: 2015, Information technology- MPEG systems technologies—Part 7: Common encryption in ISO base media file format.

[13] ISO/IEC 23001-9: 2014 Information technology- MPEG systems technologies—Part 9: Common Encruption of MPEG-2 transport streams.

[14] "Proposed Requirements for Common Media Format for Segmented Media" M37756 MPEG 114 Meeting, San Diego USA. February 2016.