A Guide To Adaptive Bit Rate Encoding

ABR explained along with how to use automation to increase efficiency

"ABR impacts not only the volume of material to be processed but also the media processing worklflow itself."

Contents

Introduction

It's often said that change is disruptive, and that's certainly true for the most significant media distribution development in recent memory: consumers moving from a "push" delivery system where they consumed media at a time designated by the distributor on a single device (the TV) to a "pull" environment where they can watch whatever material they choose, at a time that they choose, on a mixture of platforms that they own. Indeed, many viewers now switch from display system to display system as they go through their daily activities – perhaps even switching mid-program. Recent studies project that by 2021, video will consume 82% of consumer internet traffic, and 78% of all mobile data traffic. This video will come from a mixture of social media sources and curated, professionally produced material.

With this extension of video viewing from standard TVs to smart phones, tablets, connected TVs, DVRs, and game consoles, today's media landscape is radically different from just a few years ago. Consumers now expect to receive the same high-quality viewing experience, no matter what the viewing platform may be – and those platforms may have widely varying requirements. For several of the delivery options, the overriding technical hurdle is that of delivering consistently acceptable image and sound quality in the face of uncertain and fluctuating distribution bandwidths.

In the early days of multiscreen streaming, consumers often faced the dreaded "buffering" message, where a reduction in bandwidth somewhere in the pipeline caused the display device to run out of new frames to display, so playback would pause for a while. This pause would allow the player to gather enough frames to be able to start playing moving video again, but repeated pauses would destroy the viewing experience. A second issue with the state of the art at the time was that some firewalls would block delivery of the media completely, as that delivery was based on transfer protocols that were proprietary to the delivery service itself. Modern delivery of media uses a technique that resolves both of these issues – Adaptive Bit Rate (ABR) streaming.

This paper provides an overview of how ABR files differ from legacy streaming files, the implications of those differences for conventional media preparation processes, and how those processes may be successfully adapted to create highly efficient, cost-effective ABR workflows. We will also examine the latest developments in file structure (CMAF etc.) and how adoption of those technologies is reducing the complexity of providing media to a multitude of devices over a multitude of delivery pipelines.

Understanding ABR

In prior schemes, once a connection is established between an end user and a media file, that file streamed at a fixed bitrate and displays at its inherent resolution (e.g. 1920x1080). To do this, the file may actually be written to the player's storage system first and retrieved from there, complicating the startup process and being prone to the buffering problem mentioned earlier. ABR streaming–the two most popular techniques for this being Apple HLS (HTTP Live Streaming) or MPEG-DASH (Dynamic Adaptive Streaming over HTTP)–instead tailor streams to the resolution of the playback device and the available bandwidth of the connection. Connect via a tablet over a strong Wi-Fi signal, and you'll get a larger picture and higher bitrate. Connect via a smart phone on your commute and you'll get a smaller (lower resolution) picture at a lower bitrate. If your connection improves as you travel, the system will adapt on the fly to the changed conditions, increasing the bit rate at which the content is streamed.

To enable this flexibility, an ABR file isn't really an individual file at all, but rather a package of files. An ABR package includes a master manifest/playlist file, which contains the URLs (locations) of all of the metadata files for each rendition of the asset. Those metadata files are also lists of URLs, but these are the locations of the segments for a specific rendition (bit rate variant) of the asset. To enable switching between layers as conditions change during streaming, the content for each layer is fragmented into files of only a few seconds in duration (see Figure.1).

The files are constructed in this fashion to allow the player to decide which rendition it would like to use for the delivery of the next segment. The player monitors its cache of frames to see if the bandwidth of the delivery connection has increased or decreased. If it has decreased, then the player will "step down" to the next lower bitrate rendition for its next segment. If it has increased, the player can step back up again. In this way, the player can keep video playing back at normal frame rate, regardless of fluctuations in the delivery pipeline (see Figure.2).

A second major benefit of this methodology is that the data transactions between the server and the player happen purely through the use of standard "web requests", thereby removing the firewall issues other delivery mechanisms encountered.

Both HLS and DASH adopt this basic strategy, but there are some fundamental differences in how they construct the asset, starting with the way that the actual media (video, audio, etc.) is packaged up for delivery. DASH encapsulates its media into an ISO Base Media File Format – otherwise known as "ISOBMFF" – while HLS has traditionally used the MPEG-2 transport stream container. In addition, the two delivery formats use different formats for their manifest/playlist. These facts alone complicate the process of distributing media to all platforms, as it means that a distributor must create 2 identical sets of media: one for DASH and another for HLS. A further complication is that different delivery pipelines use different DRM schemes to protect the data. As you can imagine, the number of variants that a distributor must create for ubiquitous delivery is quite large and cumbersome – but hope is on the way: we'll discuss the Common Media Application format ("CMAF") and how it helps simplify workflows later in this paper.

Figure 1:
Construction of an ABR asset

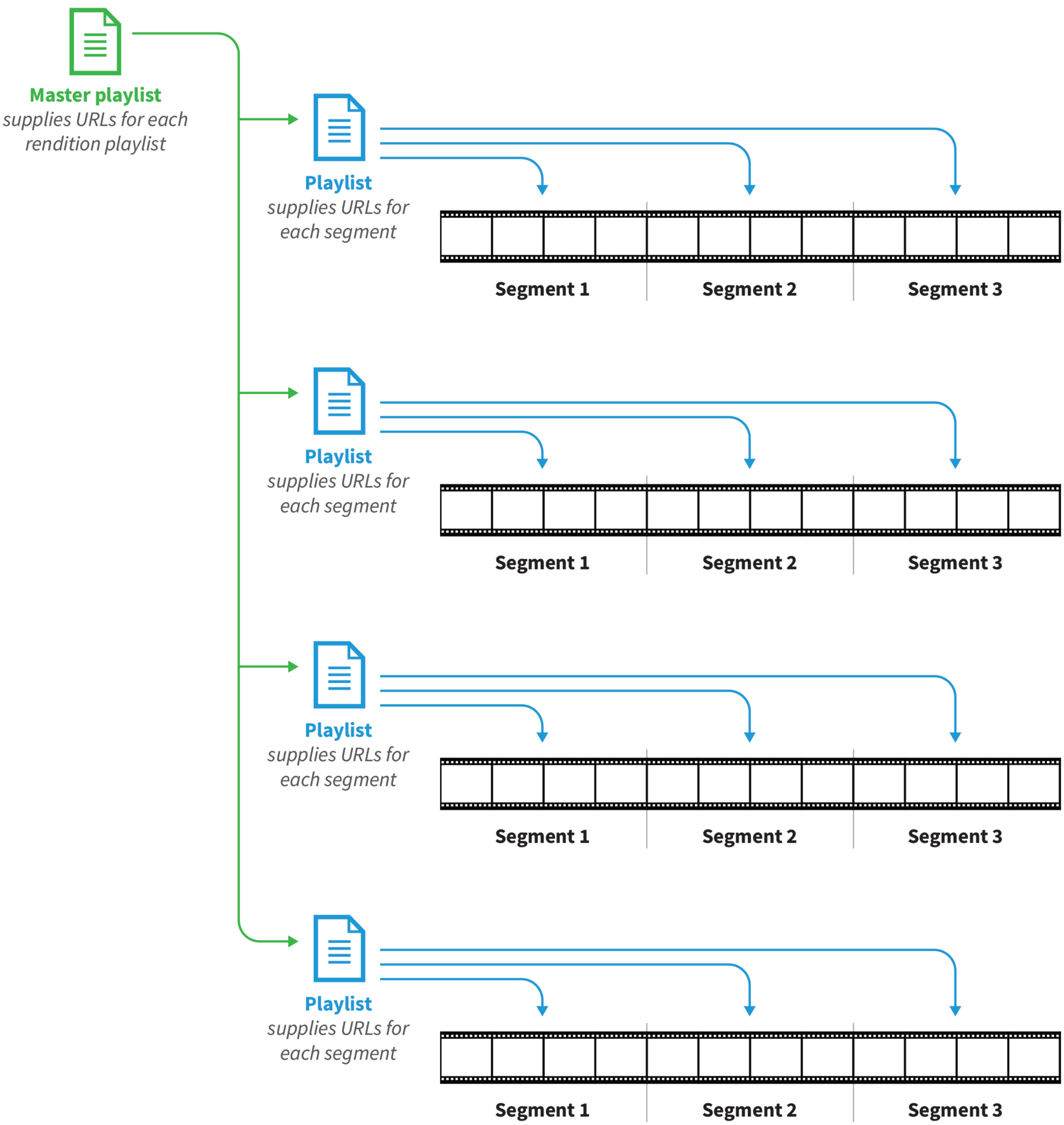
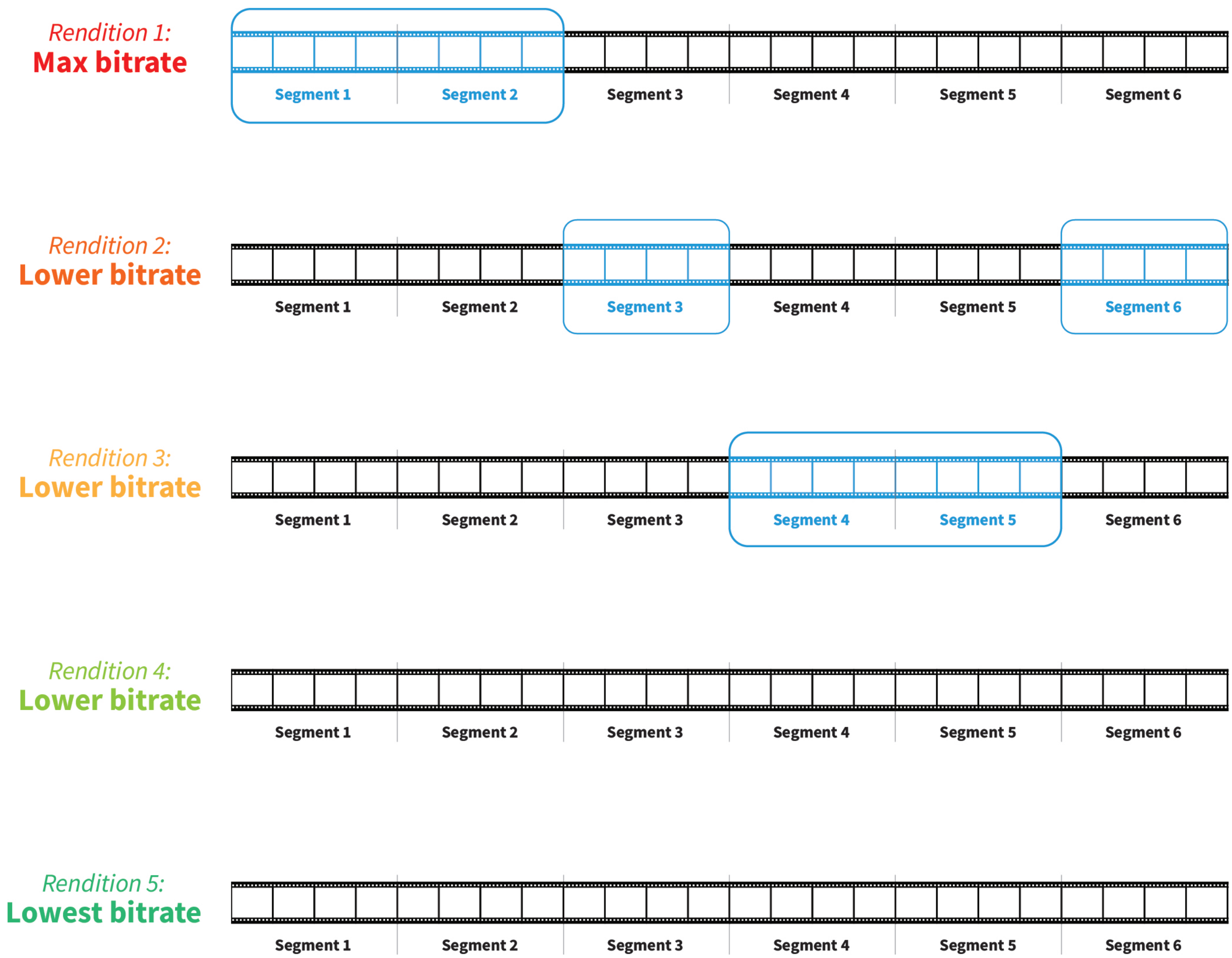


Figure 2:
The player selects the next segment to be delivered based on available bandwidth



Obviously, one major aspect of preparing an ABR package for delivery – be it HLS or DASH – is to create and format the half-dozen or so streams corresponding to each of the package's layers. If transcoded on traditional processing systems, however, transcoding for an ABR package with a half-dozen layers will typically take about six times longer than transcoding a single fixed-bitrate file of the same content. Few facilities are currently equipped to increase production by a factor of six (for each delivery format); attempting to do so without a significant investment in throughput will likely bring production to a standstill.

One approach to solving this problem would be to expand conventional capacity six-fold, investing in new machines to run the transcode processes, expanding data storage capacity, and adding all of the associated ongoing costs in areas such as cooling, energy, and real estate. But how would the costs associated with this expansion be recouped? ABR provides content owners with access to a new, fast-growing market, but it does not offer the multi-fold increase in revenues that would be needed to justify this massive expansion of capacity using existing models.

Clearly this aspect of an effective ABR solution requires a different approach to the transcoding software and hardware. At Telestream we've done that by developing a family of platforms under the name "Lightspeed", that implement parallel processing and transcoding algorithms to accelerate video processing and H.264 encoding on parallel GPUs and also on multicore CPUs. The result is the highest possible image quality at the fastest possible speed. Solutions incorporating Lightspeed provide the boost in processing power required to address the ABR throughput dilemma, but they do so without a corresponding increase in hardware, operational, and maintenance costs.

Media processing workflow

While highly efficient transcoding technology is a big part of the ABR solution, it's far from the only factor to consider in planning for ABR. That's because ABR impacts not only the volume of material to be processed but also the media processing workflow itself, requiring a thorough rethinking of existing practices. ABR emergence coincides with the significant investment that content owners and distributors have made in recent years to extend media distribution beyond the confines of traditional broadcast and cable television.

It would be a mistake to think of the content preparation side of that investment solely in terms of discrete devices for transcoding video into the correct format for various outlets. Instead, high-volume facilities have long realized that efficient content preparation demands a comprehensive approach that addresses the entire series of steps required to generate media in the appropriate form for its intended use.

Here's a look at the capabilities required of an effective media preparation system for high-volume use:

- Playlist processing – Material for a given output clip is often drawn from multiple source clips (e.g. provider logo, main content, provider promo). A content preparation system should be able to automate this assembly process, working from a playlist that specifies the exact content (source files, offsets, durations, etc.) that goes into each finished clip.
- Transcoding – The most efficient use of resources is to access source materials just once, transcoding simultaneously into all of the different required variants (progressive download, TV/VOD distribution, etc.). Transcoding capabilities must include not only video but also audio, captioning and metadata to ensure compliant results in the destination format. Built-in analysis tools should be included to provide process feedback and a means of validating transcoding outcomes.
- Content packaging – Packaging requires assembly of compliant transcoded and externally provided components into packages that are themselves compliant with requirements for delivery to targeted platforms and destinations. Because requirements vary greatly depending on a host of factors including playback platform, region, and delivery channel, a content preparation system must be able to address the entire spectrum of practices and preferences. To avoid wasteful repetition of transcoding operations, these packaging variations must be handled with zero dependency on the transcoding sub-system.
- Encryption and Digital Rights Management (DRM) – Keeping content safe from unauthorized use is crucial not only for the final deliverable but throughout the distribution chain. For an efficient high-volume system, content protection should be integrated directly into the workflow.
- Validation and tracking (QC) – In addition to ensuring the best possible rendering of transcoded media in destination formats, a system should provide the means to track the progress of a given job through the workflow and also to verify the quality and compliance of the final result before the materials are handed off.

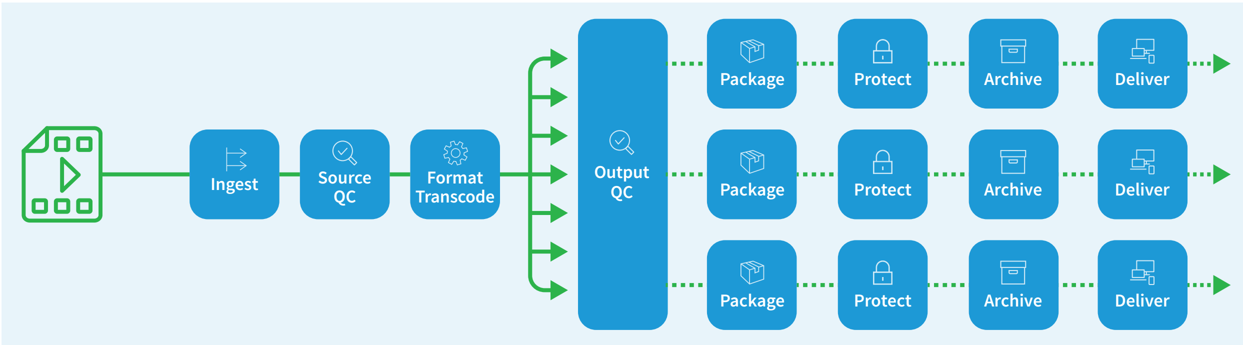


Figure 3. Unified system workflow features regain efficiencies

- Content delivery – A true end-to-end solution delivers prepared media either to a destination such as a content delivery network (CDN) or to a hosted origin server, and it should be able to confirm that delivery has been successful. With such a complex overall process it's evident that intelligent automation–with manual intervention limited to those aspects that can benefit from human judgment –can greatly boost efficiency. Any task that can only be done manually, or that must be done more than once, is an obstacle to maximum productivity. So the imperative for vendors serving high-volume content providers has been to design these inefficiencies out of the picture, maximizing quality, throughput, and control while minimizing labor. Telestream's Vantage systems, which bring transcoding, media capture, metadata processing, and analysis together into a single managed process, are a prime example of this approach. See figure 3.

Integration vs. separation

Given the tasks required of a content preparation system, and the intelligent automation needed to operate such a system efficiently, two main options present themselves for adding large-scale ABR capabilities. One is to integrate ABR into existing systems. The other is to handle ABR as a separate process. Clearly the former makes far more sense than the latter:

- In most situations, the same content will be processed into both ABR and non-ABR outputs. As noted above, industry experience has shown that when transcoding for multiple outputs it is faster to access a given source file just once and to transcode in parallel than to access the source multiple times to perform separate serial transcodes.
- Most of the other (non-transcoding) steps in the workflow will also apply to both ABR and non-ABR outputs. Again, it's more efficient to perform these steps once rather than to perform them for non-ABR deliverables and again for ABR.
- Most of the technology required to perform the needed tasks, to automate the workflow, and to track jobs through the process is the same for both ABR and non-ABR content.
- Purchasing, operating, and maintaining separate systems to handle these tasks for ABR is inherently less cost-effective than adapting existing systems and scaling them to meet combined ABR/ non-ABR demand.

Based on the above, using a unified system for both ABR and non-ABR output isn't just a nice idea, but a crucial requirement for any enterprise that intends to serve the ABR market without breaking the bank. But adapting a non-ABR system to accommodate ABR can be a tricky proposition on several levels. To make it work, a unified system must be built not only with the power to handle increased throughput but also with flexibility to handle the unique challenges posed by ABR packages. That's because, as noted earlier, the structure of an ABR file is dramatically different from that of a fixed bitrate file. And those differences can complicate several aspects of the overall file preparation workflow. Consider, for example, the task of handing off an ABR package to a content delivery network (CDN) using today's available delivery tools (FTP, file copy, etc.), which are all based on the assumption that each item of media content is represented by a single file.

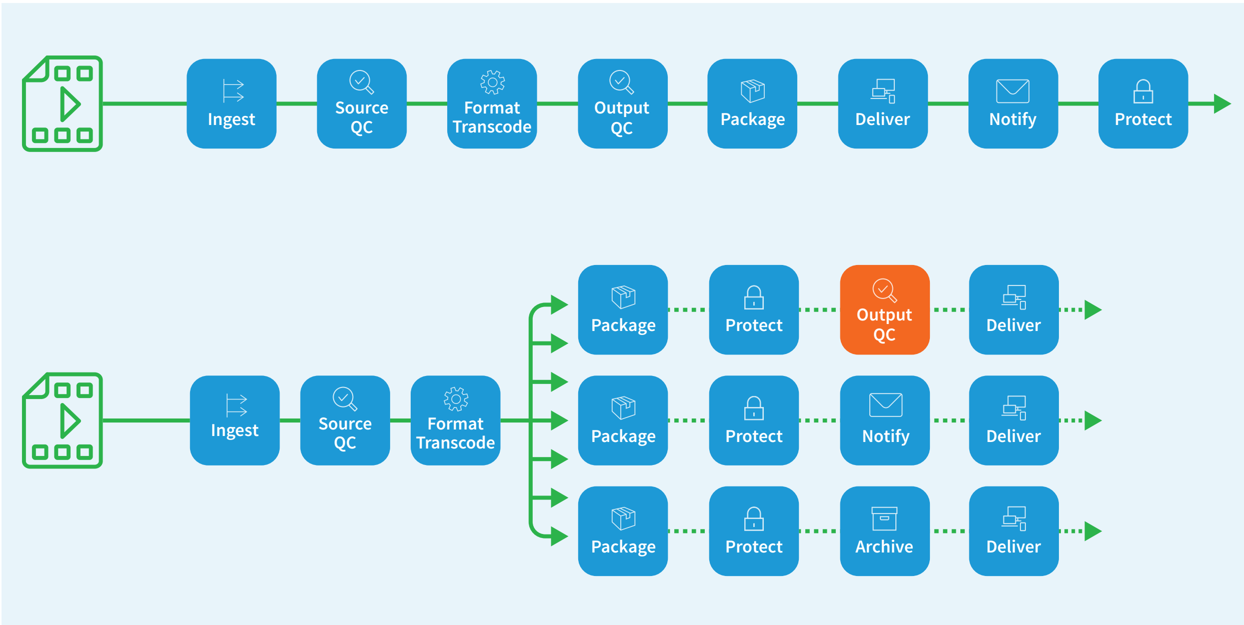


Figure 4. Mixed mode workflows

In a manual, step-by-step workflow, one could conceivably wrap the package's files into a .zip or .tar archive, FTP it to the CDN, and then rely on the CDN to properly extract the files and handle them so they function as the intended ABR package. But when demand requires production and delivery of up to hundreds or thousands of files a week –the level at which effective process automation is a business necessity - the gross inefficiency of this approach becomes immediately obvious.

This same issue plays out in many other stages of ABR package preparation. Just as for non-ABR media, the material for a given output clip is often drawn from multiple source clips (e.g. provider logo, main content, provider promo). The transcoded files must be QCed. The components, both transcoded and externally provided, that make up the deliverable must be assembled into format-compliant packages for delivery to target destinations, in some cases with DRM or other encryption applied.

An effective unified solution must address each of these stages as part of a complete automated process, capitalizing on those areas where ABR and non-ABR processing can be the same while optimizing areas that must be different.

Handling ABR with a system that does not support all of these needed capabilities is not a viable high-volume option. Vantage combines both of these into a unified system that streamlines the entire process of source file decoding, video processing, parallelized H.264 encoding, packaging, encryption, quality control, and delivery. Combining Telestream's industry-leading expertise in workflow automation and management with the power of Lightspeed technology, Vantage is the ultimate high-throughput solution to the challenge of content delivery for multiple screens.

Recent developments – CMAF and CENC

IIn 2016, Apple announced that moving forward from that date, HLS would support fragmented MP4 ("fMP4") files. To be specific, Apple agreed to support CMAF (Common Media Application Format). This is very significant, as CMAF is based on the same ISOBMFF specification that is used in DASH implementations. Finally, we have a solution to one of the thornier issues in ABR – the need to encode (and package) media twice in order to cover both HLS and DASH deliverables. We're not completely out of the woods yet, though, as the manifest/playlist disparity still exists between the two camps, but making 2 sets of manifests is a far lighter task than creating 2 sets of media, so we're definitely making big improvements in the state of the art.

One other major issue that still remains is that of DRM. Not surprisingly, DRM support is mandated by all major content providers – they want to protect their programming, along with its monetary value, from pirates. The DRM technology is included as part of the browser itself, and each browser supports a different DRM scheme from its peers: Safari supports FairPlay, Chrome supports Widevine, Internet Explorer and Edge support PlayReady and Firefox supports both Widevine and Adobe Access. So once again, distributors have to make multiple versions of the asset in order for the asset to be played back on any specific browser. There is a potential solution for this dilemma too, through the use of the Common Encryption Scheme (CENC), which specifies standard encryption and key mapping methods that can enable decryption of the same file when encrypted using different DRM systems. The scheme operates by defining a common format for the encryption related metadata necessary to decrypt the protected streams, yet leaves the details of rights mappings, key acquisition and storage, DRM compliance rules, etc. up to the DRM system itself. As of this writing, neither Apple, Microsoft of Google have made any public commitments on when they will converge on a single DRM format, so this remains a work in progress.

Recognizing the importance of these advances, Telestream products already support both CMAF and CENC.

Manifests don't have to be static!

Clearly, the accuracy of the manifests is paramount in obtaining clean playout of ABR material. They are the means by which the player can request the next segment(s) in the asset, and fundamentally enable the switching between renditions as available bandwidth to the player fluctuates over time. This might lead you to believe that manifests are static items – that they do not change over time. This is not the case, however.

One well established use case for ABR delivery is in live streaming of events. Clearly, this is an example where the manifests are being updated as the event takes place (and therefore while the players are actively playing back material.

The manifests are therefore growing during the live event. It is feasible that segments might be deleted from both the web server and the manifest as they "age out" – thereby controlling the amount of space needed for the asset (once deleted from the server, though, their entries MUST be deleted from the relevant manifests also). This approach has the added benefit of providing a means to drive a customer to an SVOD service if they wish to re-watch the event, re-monetizing the asset.

A second reason for manipulating the manifests is for local, or even dynamic ad insertion/replacement. Rather than switching the player to a different URL when ad breaks come up – which is easily detected and blocked by ad blocking software – the "local" insertion can happen at the point of distribution, simply by pointing a player to a different set of manifests for the program being watched in one location vs another. There is a momentum in the industry towards moving this manifest manipulation all the way out to the edge of the delivery network, so that "hyper-local" ad placement can take place at run time. At this point, it remains to be seen if this will, in fact, be embraced by the industry as a whole.

In summary, ABR delivery requires a comprehensive, unified system

To make an end-to-end system work effectively requires deep expertise, not simply in transcoding but also in automated media production workflows - including source-file playlists, job tracking, status reporting, and the handoff of output materials to external systems. That's precisely the expertise that Telestream has applied in developing its ABR-capable Vantage solutions. We've taken our advanced field-proven systems, analyzed the impact of ABR on every step, and created unified solutions that maintain all the advantages of full-featured video transcoding, workflow automation, and system management while also being uniquely adept at simultaneously handling ABR and non-ABR outputs, ensuring the highest quality media experience for the end user, thereby maximizing customer retention and, ultimately, revenue.

For more information on Telestream's Vantage solutions, please visit www.telestream.net