CDN Offload via Hybrid Delivery over ATSC 3.0 for Live Video Streaming

Liam Power
ONE Media Technologies
Hunt Valley, Md., United States
lpower@sbgtv.com

Abstract – One of the promises of ATSC 3.0 has been the potential for data offload. Separately, advances have been made in hybrid and IP channel rollouts on ATSC 3.0 over the last year. The two have been combined to architect and implement a data offload system. This paper explores a practical hybrid delivery model for streaming video services, allowing the distribution of video over CDNs and simultaneously over 3.0, drastically decreasing the bandwidth needs of these CDNs in 3.0 markets, all while integrating into third-party streaming apps to enable seamless streaming with no change to the viewer experience. Topics covered will include the methodology for synchronization, encoding needs for the CDN and airchain, signaling design, integration into a streaming application, and the results of the real-world testing of this system.

Introduction

As broadband video streaming has increased over the past decade, so has the bandwidth need. This presents a problem that can be solved by applying technologies in the ATSC 3.0 standard. Modern streaming services often utilize DASH for their delivery, which provides a convergence point with ATSC 3.0 that has DASH over ROUTE as one of two options for video transport, the other being MMTP. To ease the substantial burden on the content distribution system writ large, broadcasters can design a system to send the very same segments being encoded for the broadband-delivered OTT service over the air instead, delivered to the receiver with no change compared to what would arrive had they been sent over the internet. In turn, the receiver can treat the DASH segments as just another representation in the DASH ladder, seamlessly moving up and down depending on reception quality. While technically impressive, it also significantly benefits delivery efficiency, drastically reducing CDN costs for the stream.

This goal led to a set of requirements;

- **Offload:** The system must substantially improve over traditional broadband delivery for traffic usage.
- **Seamless:** The system must not negatively impact the viewer's experience.
- **Low Impact:** The system must not require dramatic modification of the streamer's DASH workflow.
- **Multi-Market:** The system must handle distribution across multiple markets to maximize offload.

Naturally, new technology comes with new challenges, and so this paper seeks to examine the various issues encountered and how they were dealt with while providing context and plans for this development:

- **Section 1** describes the problem with the current streaming infrastructure.
- **Section 2** notes some related work to address these issues.

This paper is excerpted from the Proceedings of the 2024 NAB Broadcast Engineering and Information Technology (BEIT) Conference, © 2024, National Association of Broadcasters, 1 M Street SE, Washington, DC 20003 USA.

- **Section 3** outlines the design of the overall system.
- **Section 4** illustrates a variety of challenges encountered in the process.
- **Section 5** covers the real-world demonstration conducted to prove out the system.
- **Section 6** concludes with the next steps needed for the further development.

Section 1: The Broadband Problem

As more and more households start to use streaming services, with around 90% of households using at least one Subscription Video On Demand (SVOD) service[1], and others using free ad-supported services like YouTube, FAST channels, or Advertising Video On Demand (AVOD), streaming video bandwidth usage has correspondingly increased. In 2022, over 65% of internet traffic was purely streaming video[2], with projections that video will reach over 80% shortly. Netflix alone uses 15% of internet bandwidth worldwide.[2] ISPs have consequently had to adjust their infrastructure to adapt to this drastic increase in traffic, a nearly 60x increase since 2007 when the iPhone was unveiled and Netflix launched their streaming service.[3] Current estimates are that that rate will continue to increase at 25% per year. [4]

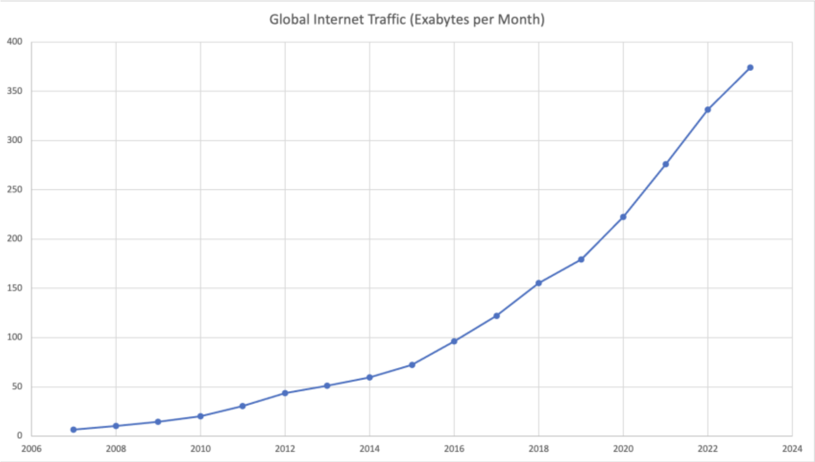

FIGURE 1: THE INCREASE IN AVERAGE MONTHLY INTERNET TRAFFIC BY YEAR.

The data usage has also impacted the streaming services. Video streaming has a tremendous infrastructure cost, forcing compromises and mitigations. On the mitigation side, there are efforts like Netflix's Open Connect project. This is a system whereby Netflix has distributed caching servers to thousands of ISP headends in almost every country on the planet at the cost of about $1 billion [5]. This allows them to store their VOD content locally at the ISP, serving it to the ISP's customers from inside the network instead of requiring the ISP to transfer the traffic from Netflix's data centers across a Tier 1 or Tier 2 network. While this is an ingenious solution for VOD with a noticeable impact on their distribution costs, it still puts them substantially above the equivalent cost for broadcast and incurs serious delay issues for live events. It also adds significant complexity to the delivery apparatus compared to broadcast, causing issues such as the “Love is Blind” live reunion fiasco in 2023 due to technical bugs (hour-long delays caused by an overwhelmed distribution infrastructure). In a notable example of the impact of this cost, the live streaming company, Twitch, recently pulled out of the South Korean market due to the high costs incurred from the 'sender-party-pays' model of bandwidth usage required in the country. This same model has been pitched in the EU and US and could further drive up costs if implemented.

Meanwhile, compromises abound for broadband – one-to-one – video delivery. Every extra bit delivered over broadband is multiplied by the number of viewers, so there is a solid incentive to compress the bitrates as much as possible. A well-known example of this is the *Game of Thrones* episode "The Long Night." This was an event viewed live by millions of people from one of the most popular shows of the time, so any increase in bitrate would have incurred a significant increase in distribution charges. Unfortunately, this was an optimally lousy episode for the highly compressed HBO streaming services, featuring very dark scenes and plenty of high-motion action. A low-bitrate video with 8-bit (non-HDR) color will suffer under these circumstances, and it suffered indeed. The telecast led to a remarkable number of complaints regarding the viewability of the episode, as many viewers were unable to discern the detail in the blotchy pictures. While HBO does not officially disclose these numbers, viewers reported a video bitrate of 3-5mbps. Even in the current ATSC 3.0 deployment scenario where 3.0 transmitters are forced to host all the main channels simultaneously, each program stream is targeted at bitrates higher than 3mbps because stations do not have the same concerns.

While the business of broadcast and broadband delivery is well beyond the scope of this paper, one thing is clear: projections for profit margins on streaming services remain well below the current margins of broadcast and cable-delivered services. In fact, every major streaming service other than Netflix is not even profitable at present. Using Netflix as an example, which likely has some of the lowest CDN costs due to its immense internal buildout efforts, the service spends an estimated $4 per user on non-content non-marketing costs like CDNs and technology.[6] When compared to the cost incurred for distribution via broadcast and cable, it is quite the leap. Most other streaming services use third-party CDNs like Amazon Web Services (AWS), which costs significantly more. Comcast's Peacock service, for example, consumed about 30 percent of U.S. internet traffic to supply the video to its 23 million viewers for the Peacock-exclusive Chiefs-Dolphins NFL playoff game this year. Assuming 5 mbps per viewer, watching for 3 hours each, that would be nearly $8 million in data transfer costs on Amazon's S3. Naturally, not everyone watched for 3 hours, and NBC presumably has a lower negotiated rate, but it provides a rough ballpark on the distribution costs for a single live event with fewer viewers than an equivalent event has on broadcast.

To highlight the efficiency disparity between one-to-one vs. one-to-many more directly, there are about 300 million cell phones in America and another 150 million tablets/laptops. If you wanted to stream the *Oppenheimer* movie to all those devices, for example, it would require about 7GB of data per device. That's ***3,150 million GB*** of data for the 1-to-1 delivery to all those devices. Contrast that with the over-the-air broadcast system that serves 210 Nielsen markets in the country, where only 1470 GB would be needed to serve the same territory. That's like turning a 30 MPG car into one getting 64 million MPG – with the ability to drive around the globe 2,600 times. That's efficiency.

Commented [GU1]: Do we really think that this movie occupies only 1GB of data?

What this in turn allows us to do is consistently deliver high quality video at a consistent bitrate to an infinitely scalable number of viewers. For reference, a rule of thumb on a 1.0 channel is 10mbps for the main channel at 1080i/720p. With ATSC 3.0, we can utilize HEVC, which allows us to realistically send 4K video in that same bitrate. Figure 2 depicts the average and minimum VMAF metrics for a sample set of 4K sports footage encoded at a variety of bitrates between 4 and 24 mbps using HEVC. It's important to consider not just the average, but also the minimum, as periods of significant drops in quality will stand out to viewers. Thus, the closer the minimum is to the average, the better we are. As the figure shows, that same 10mbps number allows us to achieve a reasonable quality minimum of 4k video using the 3.0 standard codecs, all at a significant drop in cost compared to the same programming over broadband.

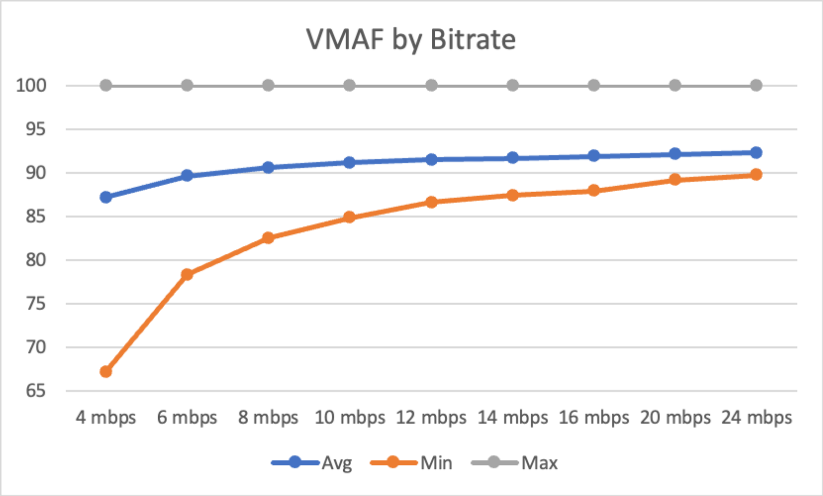

FIGURE 2: VQ RESULTS USING THE VMAF METRIC ON 4K60 SPORTS CONTENT.

All this is to say that the fundamental design of broadband presents significant issues for efficient live video delivery, as a unicast delivery model with each viewer requiring a separate stream cannot match the efficiency of a broadcast model where each viewer utilizes the same stream. How, then, can we leverage other technologies to address this inadequacy?

Section 2: Related Solutions

As a preface to a discussion on solutions, it is worth noting the development of a couple of separate but related systems utilized today.

From the broadband side, BBC R&D have been developing multicast video delivery over broadband via QUIC.[7] This would mitigate some of the above-noted issues and allow for a less hardware-intensive system of caches everywhere. It does, however, rely on the support of every ISP along the chain for full efficiency and needs quite a lot of standards work for full implementation.

Meanwhile, IP channels have been growing in usage on the broadcast side, stemming from their first demonstration by Harmonic, Sony Electronics, and Weigel Broadcasting in 2019[8]. These are channels signaled in the broadcast but with the segments delivered over broadband and potentially broadcast. Some prominent examples are WPBT in Miami, KVVU in Las Vegas, and T2 across Sinclair-operated stations. Reading this, it may be unusual to highlight that a move to broadband instead of broadcast is considered in a paper about the exact opposite. However, this is significant for two reasons.

First, this represents an example of mixing between broadband and broadcast. It does not have the same issues, methods, or benefits as the solution described in this paper, but it nevertheless opens the gates to the idea. The above can be considered a 'broadcast-first' model, where discovery is conducted on the broadcast signal, after which content is retrieved from broadband. The demonstration in 2019 used the hybrid model, in which broadband was used to provide higher quality video, start over, trick mode, and faster channel change times, while the broadcast was provided as a fallback if broadband was unavailable.

Since then, production deployments have utilized IP - "Virtual" - channels in a broadband model, with the signaling delivered via broadcast and all segments delivered via broadband. This requires the viewer to have internet access but frees up spectrum resources, allowing additional services like diginets without taking up valuable spectrum. The previous is especially important during this transition period when each market generally only has a single 3.0 transmitter due to the simulcasting requirements [9], which is generally enough to house the primary services of each station and not much else.

The latter also brings up the idea of spectrum prioritization, an idea critical to this idea. Spectrum is a finite resource, and so it makes sense to prioritize based on efficiency. A high viewership program should naturally be delivered via broadcast, while a low viewership program makes sense to deliver via broadband when higher viewership programs are in progress. Ignoring the business element, it's fundamentally a question of optimization, with the best tool being used for each job.

Section 3: System Design

One of the most significant advantages of ATSC 3.0 is its IP-based nature, with transport methods designed to work in a modern environment with multiple communication routes. This permits program distributors to engage in a dual path design where some segments are carried over broadcast while others are carried over broadband. ROUTE [10] is fundamentally another transport medium equivalent to HTTP, and DASH [11] segments are just files carried over that transport medium. The nitty gritty details are more complex of course, but the fundamental design is based around that equivalency. This implementation thus allows us to carry video over broadcast transparently and merge it with the media carried via broadband on the receiver side, allowing for a dramatically reduced CDN load, as the heaviest representation can be carried over a one-to-all delivery mechanism, which has just as much effort to send to one viewer as it does to one million viewers. The key is the ability to do that seamless delivery, which is explored here.

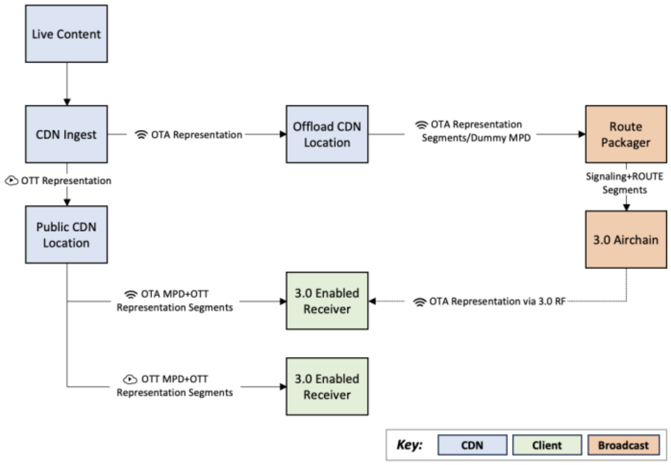

FIGURE 3: THE SYSTEM DESIGN FOR OFFLOAD, SHOWING THE MPD AND SEGMENT RETRIEVAL FOR EACH TRANSPORT.

The above diagram outlines the general system. In this, the initial stream is ingested into the CDN. The tested CDN performed its ladder encode as part of the ingest process, but this can also be done prior without any impact on the result. In either scenario, the result is that the CDN ends up with the segments for each representation. It then can generate the corresponding MPD. This system utilizes three MPDs, which will be explained in detail below.

The chosen representation(s) are ingested into the airchain, which allows the packager[1] to pull down the segments generated with minimal workflow modification on the CDN or the packager. The packager then bundles these segments into the ROUTE transport stream, along with the appropriate signaling, described later. This results in a ROUTE multicast stream with the appropriate signaling ready for transmission over the air, which can be sent on a 3.0 signal as standard, at which point it is ready for reception by a 3.0-enabled device.

Commented [GU2]: Is this a footnote or endnote?

When the viewer decides to watch the stream, the OTT process begins as normal, with the client reaching out to the streaming location to retrieve the MPD. As part of that request, however, it signals that it is a 3.0-enabled device, in the case of our PoC via query flags. Once it has done so, the server then responds with the MPD describing the OTT representations as standard, plus the OTA representation with the offloaded segments. This MPD directs the receiver to the location on the broadcast signal for said OTA representation.

The receiver then begins to tune the signal. Simultaneously, it begins playback from the OTT representation, allowing content to display immediately rather than forcing a pause for the tune time or potentially failing due to poor signal. While this content is playing, the receiver starts to receive the segments. These segments arrive through the broadcast signal exactly how they were ingested, meaning they appear with the same name and time value as the MPD describes. Thus, once the requisite signal confidence is reached, the player can cut over to the OTA representation and begin signal playback with a seamless jump to the efficient OTA-delivered segments. The logic behind this is explained in the receiver subsection below.

At this point, the OTT video is successfully offloading onto an OTA delivery mechanism, with no visible impact to the viewer besides potentially higher-quality video.

Provisioning Workflow

As part of this system, an exchange must be made between the CDN and the broadcaster, as neither knows all the puzzle pieces. The below diagram demonstrates what needs to happen to enable this:

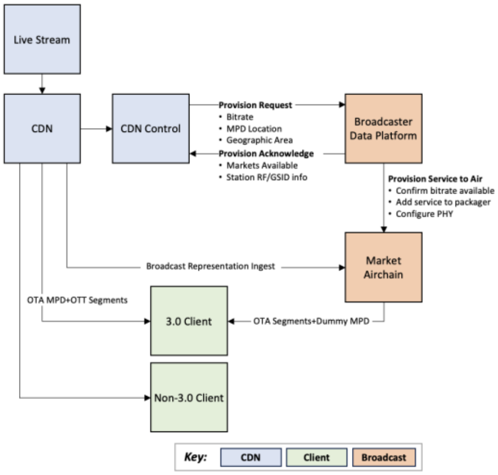

FIGURE 4: THE WORKFLOW FOR PROVISIONING AN OFFLOAD SERVICE FROM A CDN TO A BROADCASTER.

[1] Packager here refers to the device responsible for the generation of the Low Level Signaling and Service Layer Signaling, along with ROUTE and/or MMTP transport streams.

When the CDN desires to offload a stream, for example, a prominent sporting event assured of having high viewership and thus high broadcast efficiency ratios, its system contacts the broadcaster's provisioning platform. It provides the bitrate, the URL of the broadcast MPD, and the desired geographic locations for coverage. The broadcaster's platform then uses that bitrate to determine viable locations for offload. It responds to the CDN with a list of markets with available capacity, the RF channel in each of those markets to allow for geoIP bounding, and the GSID info for the service to be provisioned. The CDN then bakes this into the geolocated MPD as described above.

Simultaneously, the broadcaster provisions the services to the selected markets. This is the same methodology behind any other use of a data distribution platform, with a combination of calls to other broadcaster platforms and direct provisioning of in-house stations. These provision the segments' ingest to each packager, add the appropriate signaling, and configure the physical layer to support the necessary bitrate for transmission.

Signaling

The signaling in this system is remarkably normal for linear AV services, with minimal modifications. In the SLT [10], the service is identified as standard, with a name, ID, the GSID, and so on. One difference is that it is marked as “hidden” and "hideinguide," which prevents it from appearing unintentionally on TV interfaces, as it is a) not a complete service and b) not intended for general consumption.

In the SLS, the USBD and the S-TSID are configured as standard. The MPD contains the information allowing the receiver to identify the names and TOIs of the segments for retrieval from the ROUTE sessions, even though it will not use this MPD for playback. Note that the BaseURL is not provided in the MPD, as this is not designed for the complete product with the broadband components. In theory, the MPD could be skipped entirely, with files just sent in regular ROUTE sessions with no MPD associated, but this dramatically reduces the implementation barrier as it allows the use of existing methods.

Receiver Processing

The receiver is a crucial part of the successful display of this system, as it needs to be able to assemble the constituent parts into a cohesive whole. It also needs to be able to handle situations of no 3.0, poor 3.0, and good 3.0 coverage without displaying any disruption to the viewer experience. If it is a non-3.0 device, the process is the same as always: Reach out, retrieve the MPD, and stream the playback via broadband. If, however, it is a 3.0 device, then when it reaches out for the MPD, it will additionally have the 3.0 information described above.

Now, the receiver can use the 3.0 information to start pulling down both OTT and OTA. However, this provides no benefit to the viewer or CDN yet, as the viewer is still utilizing the broadband connection and does not see any potentially better video streams being delivered over the air. Thus, the next step is making the transition to the OTA representation.

The level of confidence required for the changeover is ultimately up to the requirements of the company utilizing the system. Given that most people will receive either all or none of the segments, with only a portion of the population in a grey area, we should reach a sufficient conclusion rapidly if the signal condition is good. At this time, the player can play OTA segments instead, thus avoiding the need to use the CDN and incur charges for distribution. After all, these segments are the very same segments generated by the encoder used in that MPD, so they should seamlessly align with the OTT representations. A generalized overview of the process of synchronization is described in the Timing subsection. Once these segments are served up via the local HTTP proxy server, the system can rewrite the BaseURL in the MPD to direct the player to reference that proxy server for the representation, and it will begin playing them out as if they had been delivered over broadband the entire time.

Timing

Timing is ultimately the most critical factor for this system. It is clear that segments can be delivered via broadcast, as that is a core component of 3.0, and segments can be delivered via broadband, as that is utilized both in 3.0 and in streaming services. The difficulty is putting those together in a live environment, where the trip time for the broadcast and broadband paths is very different due to the vagaries of CDN distribution methods, internet infrastructure, and broadcast delivery, which is part of what leads to broadcast, cable, and each streaming service having different levels of delay from true live. The delays from true live on platforms airing the 2024 Super Bowl, for example, ranged from 22 seconds for OTA at the shortest, all the way to nearly 90 seconds for the longest, the streaming service Fubo TV. [15]

One of the advantages of a segment-based system like DASH is that it is designed around switching between video streams as a foundational concept of the spec, as this allows laddering between different bitrates based on the speed of the user's connection. Each representation within an adaptation set [11] is required to be time-aligned, so switching from one representation to the next provides a seamless experience. The spec ensures that when doing so, the segments are non-overlapping and able to be concatenated.

Another critical item is the "SegmentTimeline," which provides a list of segments, named either sequentially by number or by the time that has elapsed since the start of the MPD. This allows the player to determine what segment to play to get the video for a particular time in the sequence, and vice versa.

With this knowledge, the two representations for playback can be synchronized. This would be quite simple in a world where the broadcast segment arrived at the same time that the broadband segment became available, as that is standard practice in DASH, but unfortunately, the two must be expected to arrive at different times.

Fortunately, there are several ways to compensate for this. These methods allow us to ensure the delay on all sources is matched to the player by calculating the necessary shift. These include buffering, retrieval delays, and speed alteration. Through these techniques, we’re able to ensure a predictable delay between the two. Once these are all accounted for, the sequence is able to be played out as if both were available at the same time, as demonstrated in the below diagram:

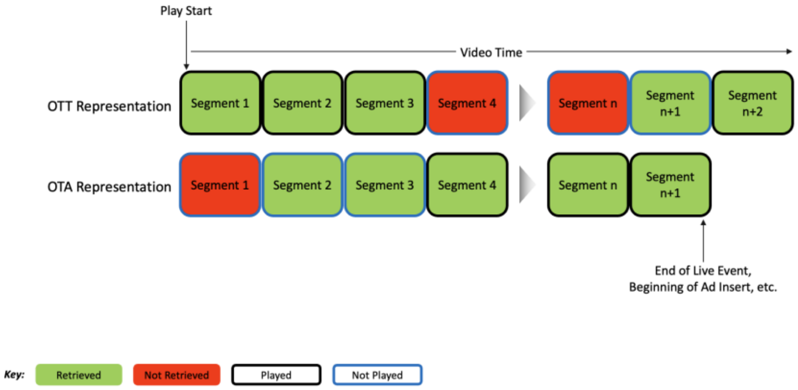

FIGURE 5: A TYPICAL SEQUENCE OF VIDEO PLAYBACK. THIS SHOWS WHICH SEGMENTS WOULD BE RETRIEVED OFF THE 3.0 SIGNAL OR THE CDN, AND WHICH SEGMENTS WOULD BE PLAYED FROM EACH REPRESENTATION.

A question that must be answered is when along the CDN process to ingest the segments. This is because CDNs generally have a pyramidal structure, where the origin is distributed to various nodes for final delivery to each geographic and network grouping of end users. What this means for the broadcast chain is that we can pull either from the end of that chain, guaranteeing that it will always be behind the CDN-delivered stream, or from the exact origin, potentially putting it closer or even ahead of the end user's CDN-delivered stream. The delay in the former case comes because, sadly, time travel has not been invented yet, and so if the air chain pulls from the same node as the end user and then adds a few seconds of delay to transit the air chain, it must therefore arrive to the end user after the broadband segments. There are pros and cons to each of these options.

Commented [SV3]: I would rewrite this to note that given the two streams are from different networks and therefore, it is expected they will not be synced. The key is make the stream predictable so that synchronization can be minimized by using technique such as bufferring, delays, etc.

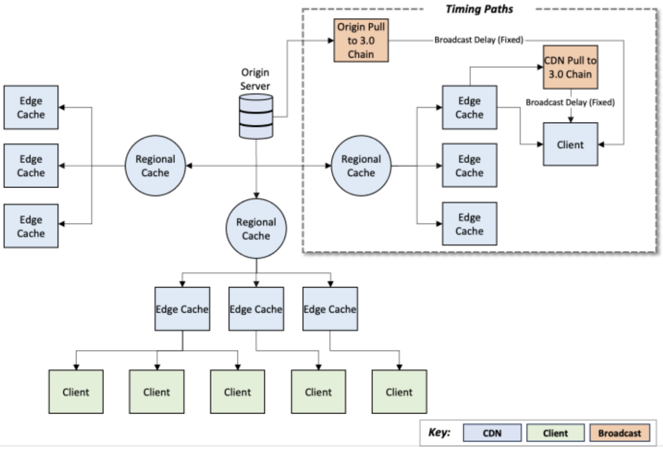

FIGURE 6: A DIAGRAM SHOWING A GENERALIZED CDN ARCHITECTURE. THE TOP RIGHT SHOWS THE TWO POSSIBLE PATHS FOR INGEST INTO THE AIRCHAIN, AND THE ACCOMPANYING TIME DIFFERENCES.

For the origin pulls, the big win is that the OTT need not be held back much, or potentially at all, to synchronize the streams. This is because the broadcast segments will be arriving around or before when the OTT segments appear in the CDN and are described in the OTT MPD, as they skip the delays of the CDN distribution process. However, this has downsides. Each packager would need to pull from the origin directly, increasing its load. This, though, can be alleviated through a central packager distribution mechanism, wherein the ROUTE packaging is done by one device, then sent out to each station using the station group's network. It also likely requires the receiver to utilize more cache, as broadcast segments are only sent once and must be stored until they are ready for playout. It also may not provide sufficient offset to eliminate the delay compared to the OTT segments.

For CDN pulls, less cache is needed, as the CDN can be relied upon to continue to have the segments available, which means it likely will need to store only one or two segments for smooth playback. However, it can incur a delay compared to the normal CDN distribution. In our real-world PoC, we found that around 6 seconds was the norm, though this varied depending on testing location and airchain configuration. It also must generally overestimate the delay between the two, as if playback of the broadband segments is started too close to current in the timeline, the broadcast will never catch up, eliminating the benefits of the offload.

Section 4: Challenges

Naturally, a system of this scale with numerous moving parts involves many challenges, ranging from differences in standard practices to features not designed with such a system in mind and even to fundamental issues with the laws of physics. The first set of issues stems from differing methods of operation, with one system designed for broadcast and the other for broadband. Segment length was the first notable one for broadcast; it was targeted to as small a segment length as possible while preserving the quality enabled by a longer GOP at the same bitrate. In most receivers, this enables a shorter tune time when using DASH-ROUTE, as the receiver does not have to wait as long to receive an entire segment. Note that most of our chunk and segment lengths are set the same.

Meanwhile, in broadband, tune time from segment length has less impact, as the receiver can download at greater than actual time speeds, which was part of the basis of the Sony Electronics/Harmonic/Weigel IP channel trial. This means that for broadband, the optimal time is often somewhat longer, as lower segment times will reduce latency and the time to scale up between representations, but at the same time, increase the risk of buffering as it catches the edge of availability and decrease the quality due to a shorter GOP length. In our testing, the CDN ended up with 2-4 seconds, rather than the 1-2 seconds usually used on broadcast-only.

Another item was the MPD format. DASH is a versatile spec with numerous features and ways of generating the MPD. A subset of these is identified in the DASH Interoperability Point for ATSC 3.0 [16], limiting the number of items our airchains generally need to be concerned with. We also generally do nothing notable with our segment locations, as there is no need when the encoder is sending them directly to the ROUTE packager. CDNs, on the other hand, gain significant advantages by using these systems, particularly those around segment location, flags for different features, and per-user dynamic MPDs, and so integrating that ingest into our various airchains proved quite challenging. With the assistance of our various packager partners, we have been able to adapt to the situation, but not without quite an effort.

Next are features. Ad insertion and trick play, or even the concept of rewinding, are items that we do not currently utilize in production. DRM, meanwhile, is used in production by other broadcasters, but not in the same way as many broadband streamers. Luckily, the former two are made quite simple with the addition of the broadband MPD, as they can be signaled as they would for an all-broadband sequence. This does mean that in the event the broadcast representation has better quality than the broadband one, the user will notice a quality drop on rewinding, or when an inserted ad plays, regardless of SSAI or CSAI based ads, but the system will still function as normal, and return to the broadcast representation once the ad is done or the user has resumed real-time playback.

DRM, on the other hand, has some additional complexity. A single-key method, as is used by those broadcasters utilizing encryption on their services, is just as functional on the broadcast representation as on the broadband one. However, when it comes to multi-key, the system breaks down. In this situation, you cannot perform multi-key encryption on the broadcast representation, as everyone receives the same broadcast segments, requiring the same key for decryption. Any in-flight encryption like this will thus result in a glorified single-key setup. This is something intended to be examined further, as we have not tested with DRM at all at present, but as it stands, we see the primary option as accepting that it must be single key.

Last is the matter of physics, which is a fancy way of saying 'timing issues.' As noted in the previous section, timing has several factors. One of the more painful issues encountered was the lack of a fixed delay number. This would have different results in different locations depending on the latency of the receiver pulling off the CDN compared to the latency of the airchain. This latency generally ranged from a four second to ten second delay between the broadband and broadcast representations. An additional confounder unlikely to be seen in the real world is the latency generated from our SRT-based STLTP distribution network for testing the same airchain in different facilities. While SRT is famously low latency, it still contributed to an increase in latency, with different behavior in each facility. Fortunately, we were able to devise several methods to adapt to this in order to handle 'on-the-fly' assessment and adjustment of latency compensation.

Section 5: Real World

Theory without practical testing is often difficult to accept, so we did exactly that. In December 2023, we conducted a real-world, on-air proof of concept of this system in Seattle, on our full power 3.0 host in the market, KUNS. Remaining within our spectrum allocation, we sent a 10mbps video representation out over the air as a hidden service with the same receivability as our regular services in that market. This 10mbps, as mentioned earlier, is what allowed us to achieve a VMAF average of 91 and minimum of 85, as seen in Figure 2. The channel did not appear on their receivers for ordinary viewers in the market, nor should it have impacted their viewing experience.

Commented [GU4]: Earlier we argue this bitrate for 4k is not a good experience. Is there context provided regarding this bitrate?

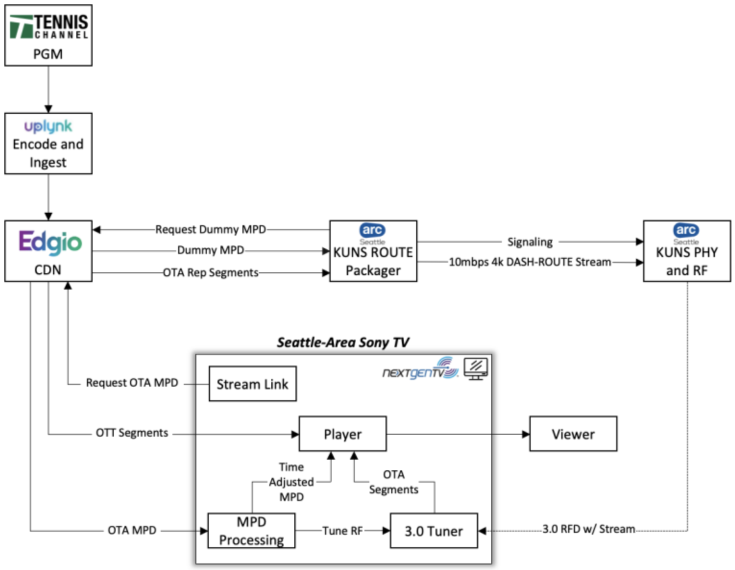

FIGURE 7: THE DEMO CONFIGURATION FOR THE SEATTLE PROOF OF CONCEPT.

This video representation was ingested as a live feed from the Tennis Channel and sent through our CDN partner Edgio's Uplynk system for encoding, distribution, and MPD generation. Edgio then configured their system to dynamically generate the three necessary MPDs, as described previously, which allowed us to ingest the top representation, the 10mbps feed, into our packager in the local market. It did the necessary signaling and ROUTE packaging and sent it through the rest of the airchain, giving us a constant feed over the air in Seattle.

Thanks to some heroic efforts by Sony Electronics to create a sample app able to handle this whole design and run on their television sets, we were able to test this on an actual TV with no additional gadgets, rather than a purpose-built receiver. This ability to run on consumer sets is vital to a production rollout, as will be explored in the next section, so we felt it was imperative to operate with an actual set from the start.

Sony Electronics went above and beyond on this effort and in an incredibly short time were able to spin up the PoC application, which pulled down the broadband MPD, began playback of the OTT stream at a chosen delay, and then tuned to the signaled channel. Once the confidence was high enough, it seamlessly switched over to the OTA representation, without a single glitch, flawlessly demonstrating the viability of this concept. We were pleased with the lip-sync component of the demonstration, wherein we sent the audio over broadband and the video over broadcast, demonstrating that the two were being played back perfectly in sync. The tennis content was beneficial here, as it is a distinctly timed sound when the ball hits the ground or a racket. Because it was the actual Tennis Channel feed, attendees could also see the same video being sent over traditional cable, showing that it was a live system.

The configurable delay was vital, allowing us to test the delay between the two for seamless failover in a real-world scenario, which is what led us to note the drastic differences in delay by geographic location, as well as the rough average of six seconds for optimal safe playback in the scenario we had configured.

Section 6: Next Steps

There are some key next steps to move this to an actual production deployment, primarily focused on receivers and streamers.

Receivers

The most critical is receiver integration. This system requires that third-party streaming apps be able to access data from a 3.0 signal, which is not a function commonly catered to in today's receivers. Indeed, most keep the hardware access to the tuner isolated to specific applications on their sets, which avoids the potential issue of conflicts over hardware control and the potential danger of that access. However, these problems can be addressed, especially given that the expectation is a few whitelisted applications rather than mass consumption of this service across the board. After all, the spectrum is limited, and there are only so many significant tentpole events occurring at a given moment.

Without this integration, we’re limited to the gateway model, using devices like the popular HDHomeRun from SiliconDust. These devices are well known amongst primary OTA viewers but less so for those who primarily consume streaming content, like most viewers of these services would be. Thus, it is better to have across-the-board implementation than to have to encourage the purchase of additional hardware in homes. The more households with receivers supporting this, the more efficient the system is compared to broadband.

Fortunately, there are clear advantages to the receivers, as this system allows streamers to provide better video content to the viewer for cheaper, encouraging increased quality. Better-looking sports and other live events are a great way to sell TVs to customers, particularly the oft advertised 4k content. Anything that enables that is a step in the right direction for everyone involved. It provides both a reason to buy TV brands supporting this system and to buy ATSC 3.0-enabled TV sets specifically.

There are two primary ways of achieving this between the two main receiver categories: direct video players, such as the various TVs and HDMI set-top boxes, and gateway devices, such as the HDHomeRun. For the former, we would need to access the tuner directly or through the TV’s middleware to retrieve the segments, which can then be utilized as described in the receiver portion of Section 3. For the latter, it opens some additional options, as we can potentially send a gateway representation that lists the local address of the gateway in the BaseURL using mDNS or similar. This would still require the timing logic on the streaming device, but it would allow us to pull down the OTA representation even on devices that lack a 3.0 tuner, such as cell phones, tablets, computers, and older televisions. Doing this would dramatically expand the impact of the offload system, as it could offset just about any usage of the live stream in the household.

Streamers

The other side of the coin is the streamer integration. The first barrier is getting streamers to use this system. Much like the receivers, there are clear benefits to doing so, as the cheaper cost of this distribution offsets the ever-rising delivery costs mentioned in Section 2. Another key point is that it is a 'broadband first' system, so anyone who lacks a device that supports this will still be able to pull the streams down as they always have. It is a supplemental benefit rather than a shift to the underlying architecture. The essential item that needs to happen for this usage is integration into their applications. A standardized TV interface for access to 3.0 signals by third-party apps would make this an attractive option, as integrating against one API is dramatically easier than integrating against a different API for every single one of the rapidly growing number of ATSC 3.0 set manufacturers. Additionally, an SDK that sits between the streaming app and the 3.0 system and handles the processing of the 3.0 input would likely help reduce the complexity of this integration, as it would keep the required additional knowledge low, abstracting it to the same DASH they are familiar with instead of the new concept of ATSC 3.0.

References

[1] Leichtman Research Group, “Emerging Video Services 2023”.

[2] Sandvine, “2023 Global Internet Phenomena Report”.

[3] Cisco, “Cisco Annual Internet Report”, 2023.

[4] Deloitte Insights, “2021 Connectivity and Mobile Trends Survey”.

[5] Greg Peters, “Mobile World Congress Keynote”, 2023.

[6] Doug Shapiro, “One Clear Casualty of the Streaming Wars: Profit,” https://dougshapiro.medium.com/one-clear-casualty-of-the-streaming-wars-profit-683304b3055d

[7] Pardue, L., Bradbury, R., and Hurst, S., “Hypertext Transfer Protocol (HTTP) over multicast QUIC,” *Internet Engineering Task Force*, draft-pardue-quic-http-mcast-11.

[8] Harmonic Inc. “Harmonic and Sony Electronics Partner to Demonstrate New SaaS Technology for ATSC 3.0 Hybrid Service Delivery at the 2019 NAB Show.”

[9] “Full power television simulcasting during the ATSC 3.0 (Next Gen TV) transition,” *Next Gen TV First Report and Order,* 47 CFR § 73.6029.

[10] *ATSC Standard:* Signaling, Delivery, Synchronization, and Error Protection, Doc. A/331:2023-02.

[11] “Information technology – Dynamic adaptive streaming over HTTP (DASH), Part 1: Media presentation description and segment formats,” *International Organization for Standardization*, ISO/IEC 23009-1:2022(E).

[12] Zigmond, D. and Vickers, M., “Uniform Resource Identifiers for Television Broadcasts,” *Internet Engineering Task Force,* RFC 2838.

[13] Berners-Lee, T., Fielding, R., and Masinter, L., “Uniform Resource Identifier (URI): Generic Syntax,” *Internet Engineering Task Force,* RFC 3986.

[14] *ATSC Recommended Practice:* Guidelines for the Physical Layer Protocol, Doc. A/327:2023-06.

[15] Kastrenakes, J., “Streaming Services are Spoiling the Super Bowl,” The Verge, https://www.theverge.com/24070601/super-bowl-streaming-delay-spoilers.

[16] “Guidelines for Implementation: DASH-IF Interoperability Point for ATSC 3.0,” DASH Industry Forum, Version 1.1.

Acknowledgments

- Graham Clift and Luke Fay at Sony Electronics, Inc. for their work on the proof-of-concept application to demonstrate the receiver side of the design
- Edgio for the work to integrate this design with their CDN
- Triveni Digital and Digicap for their rapid adjustments to their airchain products to support the necessary MPD design
- The rest of the team at ONE Media Technologies for their support, advice, information, and research to make this system a reality
- Sinclair Digital, KOMO engineering, and Sinclair operations for supporting the on-air test of this system