OTT TV for Broadcasters: The OTT encoder/transcoder-packager point of view

Introduction

As OTT TV technologies are becoming mainstream, the common feeling is that TV broadcasters could use them not only as an addition to their broadcast service, but in the near future as a complete replacement. In a March 2017 study, it appeared that 70% of respondents think that by 15 years, all or the vast majority of video will be delivered over IP instead of traditional cable, terrestrial, or satellite broadcast.

In traditional broadcast TV however, whether through satellite, terrestrial or cable, operators and subscribers enjoy a rich feature-set and a high quality of experience: high video quality itself, but also short zap times, acceptable latency, EPG, parental control, ad insertion, loudness management… To achieve the same user experience with OTT TV protocols, several technical challenges need to be solved.

This paper exposes solutions, from the OTT encoder/transcoder and packager point of view, to address those points.

This paper is an extension to “The Case for Arumai’s Private CDN for Video Customers.”

Why is OTT TV so different?

The fundamental behavioral change is that, when you want to enjoy some media content, you do not wait for it to be sent to you: you go fetch it. You do not depend anymore on a schedule decided by your broadcaster. The infrastructure of choice is of course the Internet.

What has been true for many years for off-line media contents such as music, recorded radio programs, and VOD, is now also true for live linear contents, such as what traditional TV procures.

Internet vs dedicated link

The entire point of OTT TV is to reuse the existing Internet infrastructure as the transmission link, instead of dedicated serial transmission links traditionally used by satellite, cable, or terrestrial broadcast TV. That clearly separates the roles of the broadcaster, and the carrier. What we will call “the broadcaster” in this paper is the entity offering the content as a marketed package to viewers, whether for free or for a fee, whether it creates it (produces it), or aggregates it from various sources. We will call “carrier” the entity bringing the broadcaster’s signal from the broadcaster’s place to the viewers’ places, generally via a broadband network, for instance through a delivery network. Those terms might not be the obvious choice, but we will use them to make clear that a broadcaster is what people traditionally call “a broadcast TV company”, even if technically that company does not directly broadcast the signal to viewers. Other terms could have been used, such as TV provider. Terms such as: TV network or TV station, TV channel, content aggregator, telco or MSO, all have a slightly different meaning or encompass several of those roles.

A side effect is that, as the broadcaster is now a separate entity from the carrier, the viewer is free to subscribe to any of those separately. He/she may subscribe to a carrier for Internet access only, and use it to watch free content, not having subscribed to any pay TV offer. That phenomenon called cord-cutting is explained at greater lengths in dedicated marketing studies and we're not going to detail it here.

Why use Internet as a carrier rather than anything else? Because it has many benefits:

– **Larger audience and greater reach**: as exposed above, the audience is already on the Internet. The media players are all connected to the Internet, fixed or mobile, and the viewers are searching for media content on the Internet, that they will consume on their Internet-connected devices: smartphone, tablet, PC, smart TV. In addition, Internet can be received even in areas that broadcasters cannot reach. This is especially true if we compare cable or terrestrial TV's reach to mobile Internet's3. A side effect is that it becomes realistic to create TV services that will be consumed by a number of people spread around the globe sharing a common interest in those services, instead of being concentrated in a given broadcast coverage area: typically community TV channels, or ethnic TV.

– **Lower costs**: Internet is already deployed. The broadcaster does not have to support the entire cost of creating or managing a dedicated delivery infrastructure. Instead, he uses the network access that the subscriber has already paid for. This disrupts the value chain, and broadcasters and carriers often fight on how costs and revenues shall be shared. This topic is not addressed here. It is interesting to note that delivery costs are not cancelled: broadcasters often pay third-party Content Delivery Network, whereas CDN is a free offering for licensee to Arumai's Clour Based Transcoding and Streming System for Media Companies, to overcome some of the open Internet's limitations.

Open Internet, managed networks, OTT TV, CDNs, platforms

Delivering TV services with Internet protocols does not mean the broadcaster has to use the open Internet in a totally unmanaged way. Several variations are possible.

The broadcaster may want to ensure end-to-end control of the delivery network from his place to the viewer's place. This is commonly done in IPTV. The obvious drawback is that this is not what we would call an Internet delivery: the viewers cannot access that content from any Internet connected device. It is a separate, managed network.

As opposed to IPTV, OTT TV is when the broadcaster uses a carrier that he does not control.

– He/she may choose to just expose his content on the Internet, which by essence he does not control. That unmanaged network does not guarantee QoS: reliability, bandwidth, delay, may vary.

– He often chooses a hybrid solution, involving a CDN: a specialized networking company that owns and operates network equipment and servers carefully located around the world to guarantee QoS on an as large as possible portion of the data travel over the Internet, right down to the last mile when possible. As a matter of fact, the CDN manages its own network.

– Finally, the broadcaster may also choose to use existing live streaming delivery platforms such as YouTube Live and Facebook Live, to integrate with their social networking features, and use the delivery infrastructure they already built for their own viewers[6]. That infrastructure may itself be based on CDNs, which procures the same QoS benefits as above; Arumai's Cloud-Based Transcoding and Streaming System for Media Companies ("Arumai TranStream™") is designed to manage those digital assets.

Technical consequences

This section exposes some technicalities that result from using Internet-based protocols for OTT TV.

QoS

The Internet is made of an interconnection of different service providers, and by nature a transmission from point A to point B over the Internet cannot be controlled end-to-end. Throughput, jitter, latency, cannot be as easily guaranteed as on a satellite link or a managed IPTV system. That issue is partially solved with CDNs, but still many different pieces of the connection path remain out of the carrier's control. Just think of how the viewer's perceived QoS may depend on his home WiFi gateway: "the last feet" delivery. In addition, some local network delivery protocols do not even include QoS mechanisms.

As for bandwidth, an Internet connection is usually shared between different consumers:

– shared across several receiving points: many mobile phones within a cell, many users and connected devices within a home.

– shared by several applications on a single receiving point.

That makes effective local bandwidth (throughput) hard to predict.

File-based vs streaming

Internet is efficient at delivering files: when a client navigates the Internet, his browser issues HTTP requests on port 80 to retrieve text or image data as files or portions of files. Those requests go through caches that send back the response they previously stored, or forward the request upstream, eventually reaching an origin server, only if the answer has expired or is not in cache yet.

For video streaming, a protocol such as RTMP can be used. RTMP is a streaming protocol, rather than a file-based protocol. The main drawbacks as compared to file protocols are:

– RTMP requires a persistent connection between the video viewer and the video server. That is more complex for a server to handle than a stateless file delivery mechanism (redundancy, load balancing).

– RTMP does not natively support bitrate adaptation. The player/server couple would have to decide how to handle bandwidth variations. If not, the player ends up rebuffering video repeatedly.

– RTMP travels on port 1935, which can be blocked on some network infrastructures. HTTP travels on port 80, which is always open to enable regular Internet traffic.

– RTMP is associated to Flash players, whose support has been dropped in modern web browsers and mobile devices altogether.

While RTMP can remain good at other things, it is not well suited to the distribution of video to end users over unmanaged networks. That is why file-based protocols such as HDS, HLS, MPEG-DASH, all based on HTTP, have been developed for video streaming. They expose a continuous media stream as a series of temporally contiguous file chunks, and a textual description of those media chunks in a manifest file. More, the manifest file can describe alternate renditions for the same media at different bitrates, for the player to adapt to its fluctuating receiving bandwidth and provide bitrate adaptation.

Unicast vs multicast

File delivery with HTTP is by essence unicast: an HTTP client requests the data, and only that client will be delivered with that data. In effect; HTTP is based on TCP, a unicast protocol which is designed for reliable point-to-point transmission.

Delivering video with a unicast protocol is appropriate for non-linear video: generally speaking, all VOD streaming as offered by Amazon Prime, Hulu movies, Netflix, YouTube and the likes. For streaming of live events such as news or sports, and the type of linear services that broadcasters are delivering, multicast would seem more logical.

– However, unicast can be acceptable in most scenarios, since the first stage of file caching can be efficient enough to avoid the “thundering herd”[11] issue that arises when everyone wants to see the same content at the same time.

– Multicast is anyway not commonplace on the Internet.

HTTP and delivery protocols are still a subject of discussion today though:

– Some CDN vendors offer solutions where multicast is used wherever it can be used in the network, and converted back to unicast as close as possible to the viewer’s place.

– Other protocols and APIs like WebSocket or WebRTC leverage persistent connections, or peer-to- peer, to alleviate some of HTTP limitations between a client and a server. SPDY can be used to improve HTTP. QUIC rather relies on UDP to overcome TCP limitations for Internet delivery and to improve video streaming: some CDN vendors15 may enable QUIC connections between the players and their network infrastructure to reduce connection time and improve bandwidth consistency.

– For 4G mobile streaming specifically, LTE/eMBMS allows actual broadcast of data within a cell, in a reserved bandwidth. To still enable file-based protocols like HLS and DASH which are in use on the mobile receivers, the usual HTTP protocol cannot be used as it is based on TCP which requires a bidirectional connection, but FLUTE can be used instead.

Services, metadata & standardization

Broadcast TV relies on MPEG Transport Streams as its transport format. MPEG-TS provides a framework to transport video and audio media, but also a rich and extensible description to add any kind of associated metadata. ATSC and DVB, or even more local regulatory bodies specific to a country, define more precisely how subtitles, EPG, parental rating, or other important services, shall be transported in MPEG-TS.

In OTT TV, several standards compete, the most important being HLS and MPEG-DASH (Arumai's proprietary Streaming Video Protocol as Part of Arumai TranStream™ has a much earlier priority date). They both provide ways to transport the same information but with a different syntax, which requires translation from ATSC/DVB. MPEG-DASH is more detailed than HLS on some points, for instance on how to transport EPG and parental rating. HLS can be used in a pure pass-through mode: as it is based on TS, all standard TS metadata could be delivered transparently, relying on the HLS player specific implementation to interpret them. Those protocols being file-based and not stream-based, particular care shall be taken when designing how to handle dynamic updates in an end-to-end system, as explained in the next sections.

From Broadcast to OTT TV

Broadcasters initially added OTT delivery to their regular broadcast service as a necessary evil to reach new audiences. They wonder now if OTT can be used as a complete replacement for their traditional infrastructure.

That infrastructure was based on broadcast signals, directly streamed to the viewers' devices (TV sets, boxes), with a guaranteed QoS, and well-defined services, with in-band dynamic updates. As we have seen above, all those parameters dramatically change when switching to OTT TV, which is unicast, file-based, with varying QoS, and uses new protocols for metadata and services.

This section explains, from the live video encoder/transcoder and packager point of view, what shall be done to ensure the viewers will have the same experience, with the same features and the same quality, whether their broadcasters sent them their TV services via traditional broadcast means, or desire to switch them to OTT TV delivery.

Only the live video encoder/transcoder-packager is addressed in this paper.

Zap time, latency

Although they are sometimes grouped under the same term, Zap time and Latency are two very different notions:

– **Zap time**: is the delay that separates the moment when you decide to tune in to a TV service and the moment when you first see a video frame of that TV service. If you press the TV channel button on your set-top box remote control, or click on your app's start button at 8 pm, and you see the first video fame at 8 pm + 5 seconds, then zap time is 5 seconds. It can be seen as a startup time. A large Zap time degrades usability: you cannot properly "zap" anymore.

– **Latency**: is the delay that separates the actual live action and the moment when you see it. If the goal is scored at 8pm in the stadium, wall clock time, and you see it on your screen at 8pm + 30 seconds, wall clock time, then latency is 30 seconds. It is important to keep latency low, and more specifically, lower than other existing distribution means. If you see the goal at 8 pm + 30 seconds, and your neighbor sees it at 8 pm + 15 seconds thanks to his TV broadcaster/carrier, you will certainly be prone to cancel your subscription and switch to his provider. Latency is mostly a concern for live broadcasts such as news and sports.

As can be seen, you can experience a short or a high latency with a high or a short zap time. It looks the two are not correlated. Latency has something to do with how long the live data is retained in the processing flow, whereas zap time does not. Over the decades, much work has been done in broadcast TV to make zap time, and latency, as low as possible.

Can we reach the same zap time in OTT TV as in broadcast TV?

In broadcast TV (or IPTV), the player almost has no control: when the viewer zaps, the player has to tune to the new TV channel in some way (e.g. join a multicast group, in IPTV), receives the broadcast data, and then has to passively wait for that data to be decodable. This means that zap time is impacted by the frequency of Access Points in the compressed video stream (to simplify: GOP size, or IDR period), and also by the HRD buffer size, also called CPB size, or CPB delay. Extensive literature is available on how to reduce it: reducing GOP size, and reducing CPB buffer size, are possible, at the expense of video quality.

In OTT TV, the OTT player is active: when the viewer zaps, the player starts to download the file chunks it can. Here, zap time depends on how long it takes to download the chunks (file size / bandwidth), and when the player starts to decode and display it. Reducing zap time mostly depends on the player: it may elect to download chunks with smaller bitrates during startup; or start decoding before the chunk is entirely downloaded. The encoder can also help by creating smaller chunks that can be downloaded faster, even smaller than a GOP (sub-fragments).

In summary, what can be done? In OTT TV, zap time could technically be lower than for broadcast:

– The player may decide to download the new channel's file chunks and start immediately.

– The player may decide to start with smaller chunks (of a lower rendition).

– Some encoders/packagers can be configured to create sub-fragments.

Can we reach the same latency in OTT TV as in broadcast TV?

In broadcast TV, latency is impacted by CPB delay: the player has to wait for that delay to elapse before starting to decode, to ensure that it will then receive the right amount of data to be able to store it in its receiving buffer (no buffer overflow) and also have stored enough data to continuously decode and display a picture in time (no buffer underflow). Latency is also impacted by the video compression algorithms, for a large part, as explained below.

In OTT TV, latency is inherently much higher than in broadcast TV if no care is taken. This is due to the chunked nature of OTT TV protocols. Apple, for instance, used to recommend building HLS files of 10 seconds. That implies that the OTT encoder-packager holds its live encoded content for 10 seconds before writing it as one file (one chunk) to the publication server, adding a 10 second latency. More, Apple also recommends that an HLS player starts playing with not the most recently published chunk, but 2 chunks before, making latency jump to 30 seconds at a minimum. The same rationale applies for DASH, depending on chunk size (more usually 2 seconds in DASH), and with which chunk players decide to start.

In the video compression stage itself, whether for broadcast TV or OTT TV, latency can be added by any algorithms used in the computing process: ingest and input buffering if any, video decoding if needed, video pre-processing such as image denoising or filtering, deinterlacing, cropping, color correction, scaling to other frame sizes, and video encoding: look ahead, GOP pattern, etc. Inside that processing chain, the video decoding latency has a floor value which is at least equal to the input compressed stream's own CPB size. Video processing latency can be reduced at the expense of video quality, like for zap time.

In summary, what can be done?

– Some encoders can be configured to use algorithms with lower latency. If they report their instantaneous latency, the effect is immediately visible as to which settings procure the lowest latency.

– Some packagers can be configured to create shorter segments, or even sub-fragments; or publish chunks progressively as they are built, instead of all at once.

– Some players can start playing from the most recent chunk despite Apple's recommendations; or can even start playing from incomplete chunks, provided the packagers signaled them in the manifest.

Arumai's Private OTT CDN for Licensees is designed to be able to transport "sub-fragments" to decrease delivery latency, for instance even a single video frame, provided the OTT encoder/transcoder-packager knows how to produce such video chunks. Some other CDNs may also just break down HLS chunks to deliver them progressively – but the initial HLS chunk still has to be completely created by the packager before being broken down.

The OTT players' implementation also do play a role, and much work is also done on that side. One can refer for instance to the works of DASH-IF's dash.js regarding latency[17] (and also bitstream switching algorithms[18]).

High quality OTT video on the big screen

Interlacing

In traditional broadcast TV, a mix of interlaced and progressive content can be found. Video is most often encoded in SD: 480i59.94 or 576i50, or HD: 720p59.94 or 1080i50/59.94[19]. DVB-T2 broadcast is being deployed with 1080p50. ATSC3 allows frame sizes up to 2160p and some legacy interlaced formats such as 1080i.

Bottom line is, much interlaced content still exists today and needs to be delivered with a good quality. If we take the case of 1080i, it is often broadcast as is. At the reception side, the TV set's embedded video decoder or set-top box takes care of recreating the video fields in 1080i format. If the display is a CRT, it is capable of displaying that interlaced content natively; in other cases, 1080i needs to be deinterlaced before being sent to the progressive display.

Through the years, TV vendors and set-top box vendors have improved their deinterlacing algorithms to offer the best viewing experience on the big screen, even from interlaced contents, with good results.

Can we achieve the same video quality with OTT TV? Clearly two options are possible:

– The OTT TV encoder/transcoder may deinterlace the original 1080i50 feeds to create progressive 1080p25 content. Some encoders/transcoders can even recreate 1080p50 video from 1080i50 inputs, for a better result, but using more computing power and generating more bandwidth.

– Or the OTT TV encoder/transcoder may encode/transcode in interlaced format and rely on the receiver's existing deinterlacer, provided it includes one. This is the case for most broadcast legacy equipment such as TV sets and set-top boxes, but not for pure OTT portable players such as PC, tablets, smartphones. It is interesting to note that, contrarily to the common belief, HEVC also supports interlaced modes; but not many HEVC encoders or decoders implement it.

Rate control

Traditional broadcast TV is delivered over one-way serial links with a predictable throughput: 38 Mbps for a QAM link, 24 Mbps for a DVB-T link with given modulation parameters, etc. The receiver receives that flow of media data continuously, and the broadcasting chain must ensure that the receiver always has enough data to decode and display in real-time with no interruption, and that it does not receive too much data either that would overflow its internal memory. This is ensured in the video encoder, that has to control the number of bits it produces. For this, it implements rate control algorithms such as CBR, VBR, or Capped VBR to match the available bit budget. Broadcast encoder vendors compete on the quality of their rate control algorithms to ensure that the compressed video quality stays as high as possible while the bitrate constraints are enforced.

In OTT TV, delivery is made over the Internet: bandwidth is not guaranteed, it is not a one-way link (existing protocols rather download data actively), and OTT encoders provision several adaptation profiles exactly for that purpose. Constraints are relieved, and using traditional rate control algorithms is not justified anymore. Instead, OTT encoder vendors designed custom rate control algorithms that they call ABR, Capped ABR, or with other names.

With appropriate rate control algorithms, OTT TV can reach a better video quality than broadcast TV for the same bit budget.

Preserving metadata and services

Over the years, broadcasters have added a lot of metadata and value-added services that they need to preserve when they consider switching to OTT TV delivery protocols only.

Luckily, some standards exist on how to convert them from their traditional broadcast syntax to an OTT syntax. One may refer to:

– HLS “Pantos” RFC: on ad signaling, multilingual, WebVTT captioning.

– CableLabs’ SCTE 214-1: MPEG DASH for IP-Based Cable Services – Part 1: for CEA-608/708, SCTE- 35, and color signaling, for instance.

– DVB’s and ETSI’s TS 103285: MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks: for EPG and Parental Rating (“Content Programme Metadata”), for instance.

Some examples are exposed below.

EPG (Electronic Program Guide) is usually transmitted as a binary stream over a dedicated PID in ATSC or DVB. It provides valuable information to the viewers for channel selection and also PVR/nPVR programming. It can be converted to XML-based TVAnytime format in MPEG-DASH.

EAS (Emergency Alert Service) is mandatory in the US. Being carried like Closed Caption, it can be converted (actually: passed-through) the same way as subtitles, either in HLS or MPEG-DASH.

Parental Rating, as it is called in DVB, or Content Advisory, in ATSC world, is carried as a descriptor in an MPEG-TS table. It is mandatory for regulatory compliance in a growing number of countries, even for OTT TV so-called “broadcasts”. Figure 4 explains how it can be converted to MPEG-DASH, in an emsg box.

Ad signaling, blackout, SimSub20 (Simultaneous Substitution) is also mandatory in many broadcast TV scenarios. They can be grouped in a single use case in OTT TV, which is how to signal dynamic changes of live content. From the encoder/transcoder-packager point of view: they will have to receive such signals, often carried as SCTE-104 or SCTE-35, or less frequently as ESAM. The encoder shall at least start a new Access Point where a decoder can start, typically an IDR frame. It may also perform the actual content replacement / blackout, depending on the delivery scheme. The packager shall, also depending on the delivery scheme, put back the same information in its HLS or MPEG-DASH output, for the downstream delivery chain to be able to perform content substitution. This has been described in Pantos’s HLS RFC since revision 19, and MPEG-DASH also enables this through so-called “Periods”. OTT packagers may make use of such specifications.

Content protection is a solved question: both HLS and MPEG-DASH have defined ways of encrypting contents with studio-grade DRM systems, with all necessary mechanisms such as periodic key changes, license revocation, that most OTT packager and OTT player vendors implement. MPEG-DASH has reproduced the “Common Encryption” mechanism that exists in broadcast TV: with DASH-CENC, there exists a standardized way to send the same media content encrypted once (i.e. with the same key), with metadata attached so that it is decodable by several player implementations from several DRM vendors simultaneously.

Multilingual audio, Audio Description (for the visually impaired), and Radio channels (audio-only services for digital radio) are also fully supported in OTT protocols as they are in broadcast TV, with at least all the same descriptive metadata: language codes, audio type, etc.

Loudness management (CALM Act, EBU-R128) should not be a concern: the problem was usually solved upstream before the TV service was made available for broadcast, and an OTT encoder-packager is normally not concerned by implementing loudness management, as long as it transmits the already existing audio metadata, for instance Dolby audio’s “dialnorm”.

Captioning (or sub-titling, including for the hearing impaired) is a vast and complex subject, as many text- and image-based formats coexist, and need to be translated from their original broadcast form (Closed Caption, SCTE-27, DVB-Teletext, DVB-Subtitle) to formats that OTT players support: can be the same list for some players; but more commonly OTT-specific formats such as WebVTT (text) or IMSC-1 (text or image). OTT packagers compete on how well they cover those different conversions.

About notifications

In broadcast TV, events were necessarily carried in-band as there was no other way, ensuring they were immediately notified to the player. With OTT TV, this is up to the player to download data from the delivery network, and they may not be notified immediately. That may be an issue if, for instance, Parental Rating changes from 10+ to 18+ and the player takes too long to notice it shall block displaying the content. There are several ways to overcome this:

– in TS-based HLS: the events can still be sent in-band in TS chunks and the player will be notified when it reads that audio-video chunk. They could also be put in the manifest (m3u8) file, as the player is required to reload the manifest after every chunk, typically every 10 seconds. It would save bandwidth, but the player would take 10 seconds to notice the change.

– in MPEG-DASH: as in our example for Parental Rating in Figure 4, messages can also be sent in- band, in the audio-video segments themselves (emsg box). Alternatively, they could be put in the manifest (mpd) file to not send the information at every segment and save bandwidth. MPEG-DASH though is carefully designed to allow the player to avoid downloading the mpd file after start-up. If that is the case, then a specific signal can be sent in-band, in the audio-video segments, to instruct the player to reload the manifest file, and then be notified immediately (time to download the updated manifest file).

The special case of redundancy

A rock-solid redundancy is of particular importance to broadcasters.

In broadcast TV, using a one-way communication right down to the players required to provision heavily secured, or even redundant, broadcast links if possible, and add error correction codes that helped the receiver reconstitute the missing data, at the expense of bandwidth. In OTT TV, similar techniques exist for the Internet: redundant links (by Internet's nature), error checking codes, and sometimes error correction codes. But most importantly, unlike their broadcast counterparts, OTT decoders have an active role and can download their media data from other sources if their primary source is failing. That should make OTT playback more robust than traditional broadcast, as long as Internet connectivity is guaranteed.

Thus, the difficulties of redundancy are carried over to the OTT encoders and packagers, which may become the single point of failure, before their HLS or DASH content is exposed to the Internet. Although it is not a new requirement that appeared with OTT TV, the way redundancy is handled in OTT encoders radically differs from broadcast TV techniques. Vendors may implement it in different ways:

– with N+1 hot-spare redundancy, the spare unit may have enough information to resume HLS or DASH publication from where it stopped. The delivery network, and the OTT players, will not notice any change in the linear stream, but the viewers will experience missing content, the time for the hot-spare unit to start-up.

– with 1+1 active redundancy, several units can encode the same linear content at the same time, with a synchronization that is fine enough for any system downstream to read the HLS or DASH content from either encoder and switch back and forth any time seamlessly, with no loss of content. That is difficult to ensure, and the common source of synchronization must not be a single point of failure itself.

Conclusions

It is a striking symptom that NAB and IBC, the traditional major shows in our industry, where "B" stands for

Broadcast, now heavily support OTT TV technologies.

As a matter of fact, traditional TV broadcasters who have endorsed OTT TV technologies as an addition to their services to reach new audiences, wonder now if those OTT technologies and protocols can be safely used as a complete replacement for broadcast.

Arumai thinks the answer is yes.

ARUMAI TECHNOLOGIES, INC.

Arumai is the only leading, independent, pure play OTT products and solutions company in the industry today. Arumai's grounding breaking video frame manipulation techniques, proprietary streaming systems and methods, and OTT Video Suite of products make any video content universally enjoyable in high quality on any screen, by any viewer, across any network, at any time enabling a pure play OTT products and solutions company. Every day our solutions deliver millions of content streams to mobile phones/handhelds, tablets/laptops/PCs, Blu-ray Players, Game Consoles, and Smart TVs, and in every market in the world on behalf of content owners, mobile service providers, cable companies, satellite companies, telecom operators, streaming video providers – OTT products and solutions.