WHITE PAPER

Adaptive Streaming Ad Insertion

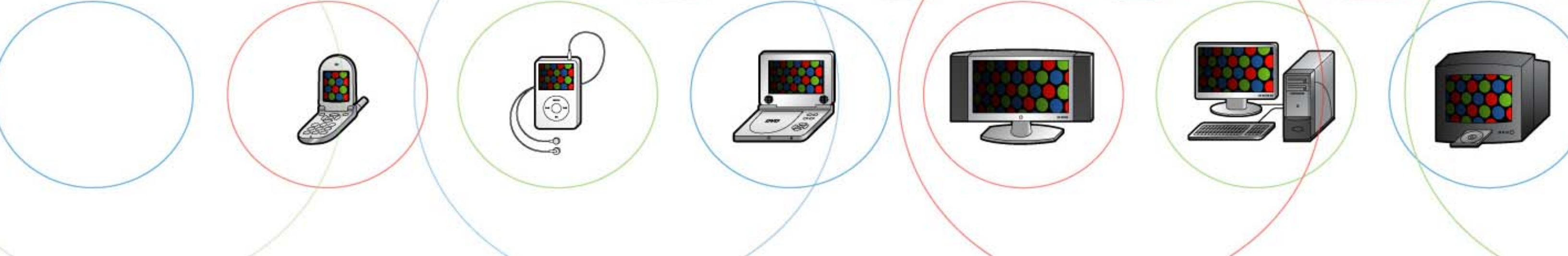
Modifying Playlists to Deliver Targeted Ads Using HTTP Adaptive Streaming

Introduction: Adaptive HTTP Streaming

Online and mobile viewing of widely-available, high-quality video content including TV programming, movies, sporting events, and news is fast becoming mainstream. Driven by the growing availability of low-cost, high-resolution desktop/laptop/tablet PCs, smart phones, internet-enabled set-top boxes and, more recently, internet-enabled TV sets, consumers have rapidly moved through the 'novelty' phase of acceptance to the expectation that any media should be available essentially on any device over any network connection. Whether regarded as a disruption for cable TV, telco or satellite TV providers – or an opportunity for service providers to extend TV services onto the web for on-demand, time-shifted and place-shifted programming environments – often referred to as multiscreen delivery or TV Everywhere – it is here to stay.

Where there is video service, there is usually advertising content to monetize the service. 'Traditional' ad insertion uses a set of technologies based on the widely used protocols for distributing UDP/IP video: ad servers, ad splicers, and an ecosystem based on zoned ad delivery. But as video delivery transport has evolved via a new set of adaptive HTTP-based delivery protocols from Apple, Microsoft and Adobe, the ad insertion ecosystem is poised to evolve to use new, targeted technologies for insertion and delivery of revenue-generating advertisements.

rgb NETWORKS



In this paper we focus on a subset of those protocols, describing how a network-centric ad insertion service would operate. The guiding principles of the infrastructure we describe are:

- The client shouldn't be aware that it's playing ads; the ads are managed in the network. This architecture gives operators more control over how ads are distributed, as well as easy access to ad viewing metrics. It also eliminates the need for proprietary clients that would be potentially required to enable ad services based on client-side ad insertion.
- Ad distribution should be content delivery network (CDN) friendly. This means that ads can be widely distributed without heavy core-network loading.
- Multiple targeting models should be possible, from zone-based down to individual targeting.
- When possible, it should enable similar services, e.g. program substitution, that utilize the same infrastructure.
- When possible, it should create new ad opportunities for service providers, in order to better monetize their video services.

Ad Insertion with Adaptive Streaming

Adaptive streaming works via client playback of a sequence of HTTP-requested short video file segments or 'chunks.' Adaptive HTTP streaming allows the client to request chunks of the same content, encoded at different bitrates and resolutions (called profiles), so that, as network bandwidth changes, the client can download the best possible quality chunk at any point in time.

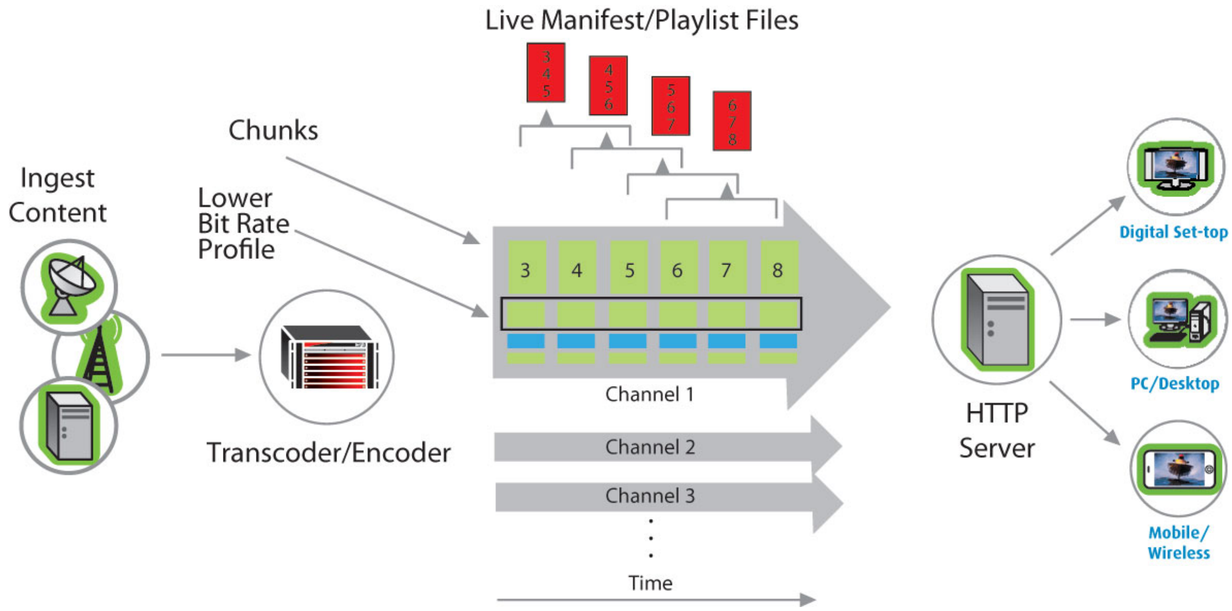
While there are three popular protocols today – Apple HTTP Live Streaming (HLS), Microsoft Silverlight Smooth Streaming (SS) and Adobe HTTP Adaptive Streaming (HDS) – Apple's technology lends itself most readily to HTTP-based ad insertion. Smooth Streaming is based on a type of 'linked list' description in which each chunk references the next chunk. That makes substituting chunks difficult, since the chunks themselves have to be modified. HDS allows chunks to be referenced via sequence numbers, but the protocol makes use of an unpublished binary format that makes manipulation difficult. In HLS, the client is notified of available chunks using a text-based playlist file that contains a list of URLs. In the case of live video, this playlist file is repeatedly downloaded by the client, so that it will have an up-to-date list of the latest available chunks. When viewing video-on-demand (VoD), the playlist file contains a reference to every chunk and is downloaded just once. For simplicity and clarity, we focus on Apple's HLS approach in this paper.

It's worth mentioning that MPEG's Dynamic Adaptive Streaming with HTTP (DASH) specification allows chunks to be referenced in a similar way (using an XML-based playlist), and it is thus also a protocol able to conveniently to use the infrastructure described below.

More details on HTTP streaming ad insertion architecture are available in a longer version of this paper, which you can obtain by contacting your RGB Networks representative.

Playlists and Ads

A simplified data flow for a live HLS stream is shown in the figure below. Different profile chunks are created at the output of the transcoder and these are referenced in the playlist file. As new chunks are available, the playlist file is updated to reference the latest-available chunks.



In order to insert ads in this dataflow, a few things must happen:

- The chunk boundaries have to align with the ad boundaries. Chunk boundaries created by the transcoder must therefore be sensitive to both the ad start and the resumption of network programming at the end of the ads.
- The playlists must be modified so that the references to chunks corresponding to ads are made to an HTTP ad server that can serve ad chunks.
- The ads have to be pre-encoded in multiple profiles and positioned on an HTTP ad server.

What Ads to Play: SCTE-130, VAST and CCMS Schedule Files

Opportunities to play ads, so called ‘avails,’ are marked using various methods, for example using a data stream specified by the SCTE-35 specification, multiplexed with the video stream. When an avail presents, the decision of which ad to play will typically depend on:

- The channel, in the case of live viewing, or the content, in the case of VoD;
- The viewer, whether targeted individually or as part of a designated market area (DMA);
- The time of day.

A few specifications have been developed to determine which ad to play.

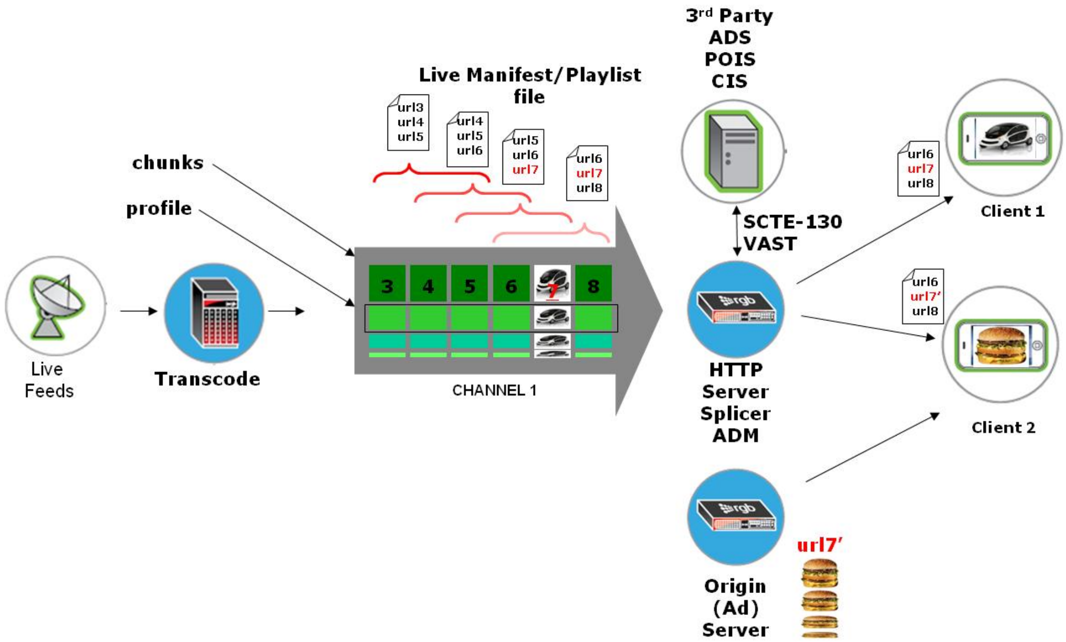
SCTE-130 specifies interfaces to multiple services that return information about which ad to play (the Ad Decision Service or ADS), the available ads (the Content Information Service or CIS), the opportunity to place an ad (the Placement Opportunity Information Service or POIS), and others. SCTE-130 can be used to target individuals or DMAs, and provides a clean interface between the Ad Management Service (ADM) that performs the ad insertion and the other components that together provide the ad campaign management.

IAB VAST was designed to provide ads in an HTTP environment and provides similar functionality to SCTE-130 using an XML over HTTP protocol. VAST is widely deployed to provide several types of ads, including interactive ads, overlays and other rich media advertising. Although it was designed primarily for VoD advertising, it can be used in both a live and VoD environment to specify which ads to play in-line in a video stream.

Lastly, legacy ad insertion is often based on CCMS schedule files which specify which ads to play, in which channels, to what zones and at what time. CCMS schedule files include enough information to provide a starter service, but they cannot scale to fully targeted ad delivery.

Ad Insertion via Playlist Manipulation

An HTTP adaptive bitrate ad insertion architecture is shown in the figure below. In this example, chunk 7 represents an ad. (note, however, that normally ads would span several chunks). The two clients receive modified playlists that refer to different chunks during the ad. Client 1 plays the network ad, but client 2 receives a modified playlist with a reference to chunk 7' which resides on a separate HTTP ad server. Note that the ad server can be a normal HTTP server. It simply must have the ad chunks pre-positioned and ready for serving. The decision of which ad to play for which client can be determined by an ADS via an SCTE-130 or VAST interface, also shown.



Program Substitution

In some situations, content rights do not allow programs to be displayed in some regions. In this case, it is convenient for service operators to substitute one program for another based on a schedule. Within the framework described above, this can be managed by treating the program as a long-form ad. The CCMS file mechanism used to identify the ad that a zone should receive can be used to determine which program the zone receives.

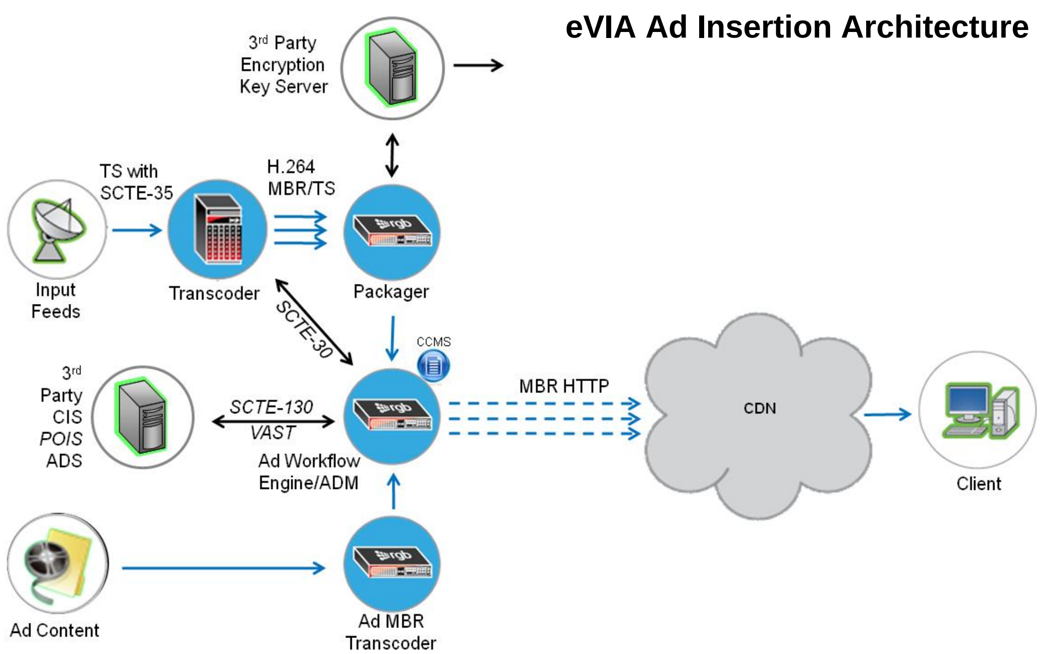
Live and VoD

Live and VoD streams have very similar workflows in HTTP streaming. The only difference is that in the live case, the playlist is modified as new chunks become available and thus the client repeatedly downloads it in order to have the most up-to-date version. In the VoD case, the playlist is created just once with all the references to every chunk included. The mechanisms used to deliver ads in live streams can be used to deliver both pre-roll, mid-roll and post-roll ads in a VoD framework.

RGB's eVIA Architecture

RGB's Enhanced Video Intelligence Architecture (eVIA) provides a suite of products that provides advanced HTTP streaming video services for delivery and monetization of video to multiple screens. The eVIA suite is designed for flexibility and scalability, with industry-leading density and high-availability designed into every component.

The eVIA ad insertion ecosystem, shown below, is dependent on a number of components with requirements specific to both adaptive streaming and ad insertion. We describe some of these here, including third party components that are an integral part of the overall ecosystem.



Ad locations in network streams are often marked with SCTE-35 cues multiplexed on the input data stream. SCTE-35 cues can signal the beginning of an ad, its duration and its ending. However, the usage of SCTE-35 is variable, and often the duration of the ad is not signaled. This means that to be completely general, the complete system must be able to determine the duration of an ad avail and report this to both the transcoder and the packager, as described below.

The Encoder/Transcoder

RGB's VMG transcoder is responsible for ingesting the content, processing the video and audio streams, and preparing it for segmentation. The VMG processes the video in the following way:

- The output video is de-interlaced, if necessary, to progressive format.
- The output video must be scaled to resolutions suitable for the client device.
- Audio must be transcoded into AAC audio, one of the audio codecs supported by Apple HLS.
- If SCTE-35 is used for ad insertion, the VMG will add instantaneous decoder refresh (IDR) frames at the ad insertion points, so that the video is ready for ad insertion.
- The different output profiles (that is, resolutions and bitrates) must be IDR-aligned so that the client playback of the chunks created from each profile is continuous and smooth.
- The VMG uses an SCTE-30 interface to the Ad Workflow Engine (AWE) that can specify the ad duration so that the transcoder can insert an IDR frame at the end of the avail (that is, at the 'splice-in' point). This allows the network programming to start at a chunk boundary as well.
- A unique fault tolerance mechanism in the VMG allows two different VMGs that ingest the same input to create identically IDR-aligned output. This can be used to create a redundant backup of encoded content in such a way that any failure of a primary VMG is seamlessly backed up by a secondary VMG.

Because the quality of experience of the client depends on having a number of different profiles, it is desirable to output a large number of different profiles for each input. Deployments may use anywhere from 4 to 16 different output profiles for each input.

RGB's VMG supports these features in a high-availability chassis with the highest capacity available in the industry today. The VMG also features built-in, redundant networking, fans, and power supplies, managed by a single management interface.

The Packager

RGB's AMS/TransAct Packager is the component (also called a segmenter or fragmentor) that takes the output of the VMG transcoder and packages the video for the specific delivery protocol. The packager supports the following features:

- Packaging capability for Apple HTTP Live Streaming (HLS).
- Encryption capability – the Packager can encrypt the outgoing chunks for HLS using file-based AES-128 encryption.
- Integration with third party key management systems – the Packager receives encryption keys from a third party Key Management Server (KMS) that is also used to manage and distribute the keys to the clients.
- Encryption key origination – the Packager can create its own encryption keys, if integration with a third party DRM vendor is not desired. The keys can be published for distribution to clients.
- The Packager can ingest live streams or files, depending on whether the work-flow is live or on-demand.

- The Packager easily interfaces with CDNs by allowing both reverse proxy connections and put scenarios via HTTP PUT or a mounted NFS/CIFS directory. In the pull use-case, the Packager includes caching headers on HTTP responses for proper CDN cache control.
- For ad insertion, the packager is SCTE-35 aware and able to create playlist files with modified URLs during ad durations.
- The Packager has an interface to the AWE that tells the Packager the duration of the ad, so that it can return references back to the main network program at the appropriate time.

RGB's TransAct Packager offers these features in both a software and hardware configuration.

The Ad Workflow Engine (AWE)/Ad Management Service (ADM)

The AWE is the control interface that manages all of the other components. It provides ADM capability for controlling the actual ad insertion. It has the following functionality:

- Ingest of ads with workflow to transcode ads and position them for delivery.
- Ingest of CCMS schedule files. The CCMS schedule file capability provides built-in Ad Decision Service (ADS), Placement Opportunity Service (POIS) and Content Information Service (CIS) for legacy, zoned ad insertion.
- Ingest of playlists from the Packager for modification and distribution.
- A SCTE-130 interface to external services that provide ADS, POIS and CIS functionality. This allows the AWE to deliver targeted ads to individual users.
- Logging service is provided by collecting and processing the logs from the other components, including ad transcoding logs, ad delivery logs, and system status logs.

The Ad Encoder

Clients may adapt bitrates during an ad, just as they might during any other part of the streamed video, and so the ads must be encoded into the multiple profiles used in adaptive HTTP streaming. This functionality is provided by the AMS/TransAct Transcoder. It supports:

- H.264 video encoding
- AAC audio encoding
- IDR-aligned TS files, as used by HLS, encoded to as many profiles as the main video program
- The ability to create different duration chunks, to match decisions about chunk duration made in the main video program

Third Party Components

Content Delivery Network (CDN) – RGB does not provide the components of a distribution network, although RGB does offer CDN origin services. RGB's components have been integrated into the infrastructure of a number of CDN providers.

The content delivery network needed for HTTP streaming is simple – it is HTTP-based and doesn't require any special streaming servers. For live delivery, it is beneficial to tune the CDN to age out older chunks rapidly, as there is no need to keep them around long. The actual duration depends on the duration of the chunks and latency in the client, but a minute is normally sufficient.

It is important to note that the number of chunks can be very large. For example, a day's worth of 2-second chunks delivered in 10 different profiles for 100 different channels creates 43 million files! Thus the CDN must be able to cope with a large number of files as well.

Key Management Server (KMS) – Although RGB's Packager encrypts the chunks for secure delivery, RGB does not offer a KMS for delivery of encryption keys to clients. RGB provides an API that can be used by any KMS to exchange keys with the Packager; API integration has already been completed with a number of partner KMS vendors.

Ad Decision Service (ADS) – While the AWE does support legacy, zone-level targeting via CCMS schedule files, RGB's eVIA system relies on external components to manage ad campaigns or target individual users. For this functionality, RGB supports standard SCTE-130 interfaces and works with third party providers of ADS, POIS and CIS functionality.

Conclusion

Online and mobile viewing of premium video content is increasingly providing a complement to the traditional TV experience. In order to monetize premium video services over new protocols, a new video ad insertion infrastructure is required, and RGB, utilizing years of ad insertion expertise, is at the forefront of bringing these newer ecosystems to the market.

eVIA from RGB brings together a suite of tools that enables service providers to monetize HTTP streaming services, whether on managed or unmanaged networks. The ability to create a new ad avail with live pre-roll may provide significant revenue opportunities that will help speed rollouts of new services. These services, combined with the flexibility of the eVIA suite to provide other services, including live and VoD transcoding, network PVR functionality, and program substitution will take HTTP streaming services well past today's capabilities with shorter adoption time than ever seen with new technologies.

rgb NETWORKS

www.rgbnetworks.com

R1105-0911-01