ADAPTIVE STREAMING – NEW APPROACHES FOR CABLE IP VIDEO DELIVERY

John Ulm, Tom du Breuil, Gary Hughes, Sean McCarthy
Motorola Corp., Home & Mobility

Abstract

Adaptive Bit Rate Streaming is a technology being deployed to deliver IP video to personal computers and mobile devices over the internet. This paper provides a tutorial on this technology and its application in Cable IP Video delivery systems.

We will explore the impact of Adaptive Bit Rate Streaming on topics such On Demand unicast services & Linear TV multicast services; transcoding; unmanaged home networks; Ad insertion; impacts on CDN; video bandwidth efficiencies and Migration Strategies.

INTRODUCTION

Interest continues to accelerate for supporting IP Video on a cable infrastructure. Many factors have contributed to this including the exponential growth of over-the-top entertainment quality video, more broadband homes with higher speeds, significantly more efficient H.264 video and AAC audio codecs and the ease of integrating PC and smart phone experiences.

One of the critical questions is how to choose the best video delivery mechanism for all IP delivery. Recently we have seen significant interest in using emerging Adaptive Bit Rate Streaming protocols from the mobile and PC arena for cable IP video delivery. Called Adaptive Streaming for short, it enables smoother playback across a variety of internet-connected devices and is optimized for internet video delivery. But how well suited is adaptive streaming for IP video delivery across all screens including the TV?

This paper will provide operators with an overview of the new adaptive streaming protocols. IP Video delivery has evolved from streaming and progressive downloads to something that's evolved to scale for world wide delivery. Key to this is the use of HTTP for the underlying video transport. Various proprietary ecosystems have been deployed by companies such as Apple, Microsoft and Adobe while various standardization efforts are now underway.

We discuss the strengths and weaknesses when using adaptive streaming for IP video delivery over a cable infra-structure. Managed IP video service must consider both Linear TV and On Demand content delivery. Linear TV is associated with real-time and often multicast delivery while On Demand is stored content with unicast delivery. A critical issue an operator must tackle is where to transcode the IP video into the various formats. Other topics include Ad insertion, delivery over unmanaged home networks, policy & asset management and finally, migration strategies.

These new protocols will also have a significant impact on an operator's Content Distribution Network (CDN), video servers, and edge distribution. Servers must evolve from their current streaming operations and efficiently handle the multiple new formats needed for each asset. We'll also take a look at Trick Mode support and CDN bandwidth and caching.

Finally we will take a look at the video bandwidth efficiencies that we gain with adaptive streaming compared to traditional video broadcast models, including the impact of Variable Bit Rate (VBR) video delivery.

OVERVIEW – ADAPTIVE STREAMING

Background

Traditional Internet Streaming video delivery to PCs was designed with real time protocols and provides the content as you need it with minimal buffering requirements. Some common protocols used include Real Time Protocol (RTP) for the video transport and Real Time Streaming Protocol (RTSP) or the Real Time Messaging Protocol (RTMP) over TCP for the control. These stateful protocols work well in controlled networks including enterprise or service provider environments.

The real time nature provides a responsive user experience with well defined bandwidth usage. However, the real time nature often requires a separate network for video streaming and doesn't work well for distribution over the internet. This approach also does not support standard CDN networks using HTTP caching and has potential scaling issues.

Traditional Streaming also has issues in traversing through firewalls in routers. The real time protocols use ports that are different from traditional web browsing and often require the router/gateway to be manually configured. This is a significant problem for wide spread use in consumer managed home networks.

Progressive Downloads were designed to deliver content over the internet. It works from a standard web server and uses HTTP as the transport protocol. This enables it to scale on a world wide basis by leveraging standard HTTP caching and it eliminates any issues with getting through firewalls since it uses the same ports used for web browsing.

However, there are several significant drawbacks to Progressive Downloads. User experience is impacted with significant latency while the buffer is filled and re-buffering followed by video pauses when there are insufficient network resources.

Progressive Downloads can place added buffering requirements on the user devices and be wasteful of bandwidth as well, especially with un-throttled download speeds. In a very common use case, a user stops watching the content after a short period of time (e.g. channel surfing), but most or all of the video content is still downloaded, consuming excessive network resources.

A Hybrid Approach – The Best of Both

In both streaming and progressive downloads, the video content is encoded with a fixed rate/quality model. If available network bandwidth is reduced, the user may experience starts and stops in the picture or be forced into long delays as buffers fill. If network bandwidth is in abundance, then the user may be viewing content at lower quality than what the system is capable of delivering. Neither protocol adapts well to changing network resources.

So the key challenge is how to deliver great viewer experiences over variable uncertain bandwidth to a wide variety of display devices, not just PCs. Adaptive streaming was developed to capture the best of both streaming and progressive downloads.

Basic Operation - Chunking and Play Lists

Adaptive streaming is a hybrid delivery method that acts like streaming but is in fact a series of short HTTP progressive downloads. It relies on HTTP as the transport protocol and performs the media download as a long series of very small files, rather than one big progressive download file.

The content is cut into many small segments and encoded into the desired formats. These small segments are often called chunks, streamlets or fragments and typically cover 2 to 10 seconds. A chunk is a small file containing a short video segment along with associated audio and other data.

Adaptive streaming typically uses HTTP as the transport for these video chunks. This gives it all the benefits of progressive download. The content can easily traverse firewalls and the system scales exceptionally well for high demand as it leverages traditional HTTP caching mechanisms.

By using small chunks of video, adaptive streaming also behaves like traditional streaming and is applicable to both live delivery and pre-stored on demand content.

The new twist that adaptive streaming introduces is the ability to switch between different encodings of the same content. This is illustrated in Figure 1. Depending on available bandwidth, you can choose the optimum encoding thus maximizing user experience.

Each chunk or fragment is its own stand-alone video segment. Inside each chunk is what MPEG refers to as a GOP (Group of Pictures) or several GOPs. The beginning of each chunk meets the requirements of a Random Access Point, including starting with an I-frame. This allows the player to easily switch between bit rates at each chunk boundary. This collection of multiple adaptive streams each with different encodings is some times referred to as an adaptive stream bouquet.

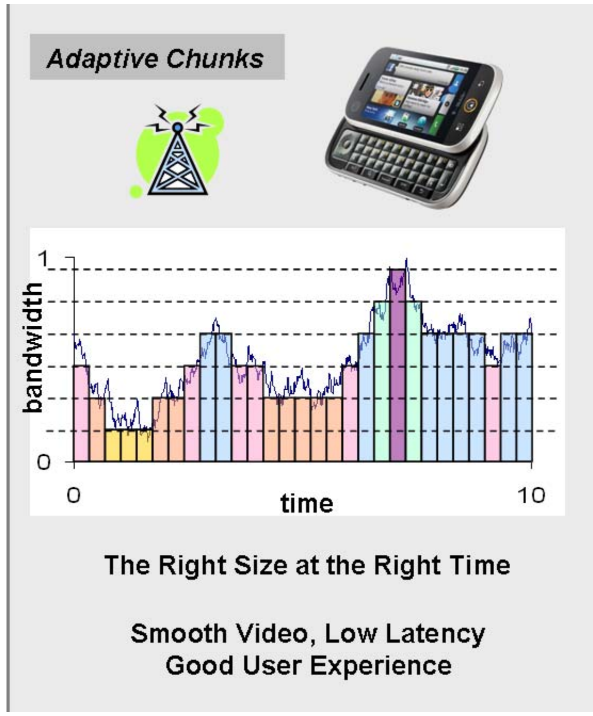

Figure 1. – Adaptive Streaming: Basics

Adaptive streaming also allows a user to start the video quickly by initially using lower bit rate chunks and then quickly switching to higher quality chunks. This provides a straightforward solution for fast channel changes, a feature valued by consumers.

Central to adaptive streaming is the mechanism for playing back multiple chunks to create a video asset. This is accomplished by creating a play list that consists of a series of URLs. Each URL requests a single HTTP chunk. The server stores several chunk sizes for each segment in time. The client predicts the available bandwidth and requests the best chunk size using the appropriate URL. Since the client is controlling when the content is requested, this is seen as a client-pull mechanism, compared to traditional streaming where the server pushes the content.

Using URLs to create the play list also enables very simple client devices using web browser-type interfaces.

Ecosystems and Standards

HTTP chunking is an underlying transport mechanism. To create an end-to-end system for video delivery requires additional components such as video and audio codecs, Digital Rights Management (DRM) and other control plane elements. As of today, different proprietary adaptive streaming ecosystems have emerged from companies including Apple Computer, Microsoft, Adobe and Move Networks.

Move Networks was an early adopter of the technology and showed in 2008 that their adaptive stream HTTP-based media delivery could be done successfully on a large scale. This included broadcast of the 2008 Democratic National Convention using Microsoft Silverlight[tm] as the client framework [1].used by several programmers for streaming their content over the internet [2].

Microsoft created a prototype HTTP-based adaptive streaming for the 2008 Beijing Summer Olympic games. However, this experience exposed the issues of managing the millions of tiny files that were created during this very large event. Microsoft then introduced Smooth Streaming to overcome these shortcomings by defining chunks as movie fragments stored in a contiguous MPEG-4 Part 12 (MP4) file, using features of the MP4 format to mark chunk boundaries for easy random access. This sub-format is referred to as a Fragmented MP4 file.

Apple refers to its Adaptive Steaming implementation as HTTP Live Streaming and it is used to deliver media to the iPhone and iPod Touch. Quicktime X can also play HTTP Live Streaming, enabling playback on the PC.

Adobe has worked on an extension to RTMP called RTMP Chunk Stream Protocol. While designed to work with RTMP, it can handle any protocol that sends a stream of messages.

All these ecosystems use Advanced Video Coding (AVC, a.k.a. MPEG-4 part 10 or H.264) for their video codec and generally use AAC audio. These modern codecs are valued for their efficiency. However, each ecosystem supports different chunk file formats, typical or recommended chunk sizes, chunk file overhead, number of files to manage at the server and ways of chunk file creation (pre-stored or on-the-fly real time).

Standardization efforts for adaptive streaming are under consideration within several standards bodies and industry groups. Several IPTV organizations are considering adaptive streaming while Apple has submitted a draft of HTTP Live Streaming to the IETF for standardization [3]. At this stage, it is too early to know which efforts will prevail and in what time frame. Some of these efforts may support more than one profile in order to interoperate with one or more of the existing proprietary adaptive streaming ecosystems.

ADAPTIVE STREAMING IN A CABLE ENVIRONMENT

Delivering across the cable managed network

Adaptive streaming uses a client-pull rather than a server-push delivery mechanism and because of this, clients can automatically and dynamically adapt to the available network bandwidth available to them, enabling a smooth video experience albeit with variable video quality. This is extremely useful for over-the-top unmanaged internet delivery of media services. As such, adaptive streaming provides excellent support for the three screen subscriber experience when they are out of their home and off their cable provider's managed network.

This leads to the question on the role or use of adaptive streaming within the cable provider's managed network. Since adaptive streaming is client driven, each viewing session is unicast and therefore needs its own bandwidth, independent of whatever other subscribers in the neighborhood may be watching concurrently.

A provider can use different approaches to manage the user experience for this environment. These broadly fall under categories of adding sufficient bandwidth, limiting delivery to select devices/subscribers or limiting which content/applications uses adaptive steaming.

One approach is to over-provision the IP network capacity to exceed the combined bandwidth requirement of all the concurrent subscriber demands in a neighborhood. In the near term cable environments, this appears impractical until DOCSIS 3.0 costs come more into line with traditional video costs such as Edge QAM devices. There is also the issue of available spectrum which might require node splits to garner sufficient additional IP bandwidth.

Another avenue is to limit the number of devices or subscribers receiving the new IP video delivery. A provider could limit the IP video service to only PC and mobile devices, or limit the service to only their premium customers. This is another way to ensure that the available IP bandwidth is sufficient for the offered IP video load.

An alternative approach for cable providers is to manage the delivery of IP content between multicast and unicast delivery. The provider can deploy popular content as IP multicast on their networks and reserve unicast for only those services that are uniquely being consumed. This is similar to Switched Digital Video, SDV, today in that services that are not currently being watched aren't transmitted. This provides two significant benefits in that the service provider can guarantee the subscriber experience by selecting their preferred quality for each multicast service while minimizing the total network bandwidth consumed for this delivery since only a single version is delivered to multiple subscribers concurrently.

As an added refinement, the unicast services can take advantage of adaptive streaming which allows the cable operator to better manage their network resources while still providing a good customer experience. This approach with adaptive streaming for unicast services provides the same benefit as SDV in today's video networks with the additional benefit of automatically allowing more simultaneous unicast sessions at lower bitrates or fewer at higher bitrates.

<u>Delivering across unmanaged home networks</u>

Today, most consumer networking equipment, WiFi, or other retail video products do not support multicast delivery. However, providers can support multicast delivery through the gateways, set tops and home networks that they install and manage. Because of this, a short term strategy for providers may be to multicast services to IP set tops in the home via a service provider managed high bandwidth home network such as MoCA and then use Adaptive Streaming unicast services over the access network to other subscriber home client devices such as

PCs or WiFi-enabled smart phones over unmanaged in-home networks.

In smaller residences in sparse neighborhoods with clean WiFi installations and limited concurrent demand on this home network, service quality may be fine. But in larger homes or locations where multiple adjacent WiFi networks are competing for the same spectrum, the end user experience may suffer the usual media interruptions and re-buffering instances that were so common in over-the-top video experience prior to the introduction of adaptive streaming.

One solution is to send the entire adaptive stream bouquet from the network such that the gateway can act as a proxy for the actual PC/phone client and can forward the requested bit rate stream from this bouquet, with the obvious drawback of consuming more of the available DOCSIS bandwidth in the cable provider's network.

Another alternative is to provide one or more real-time transcoders in the gateway that can be used to dynamically transcode the source stream to the available target bandwidth on the fly. This approach adds no overhead to the DOCSIS network, but does add the cost of the real-time transcoder in the gateway device. Note, gateways may need multiple transcoders if they are to serve multiple clients concurrently.

<u>When & Where to Transcode</u>

As we just discussed, adaptive streaming requires content in many different formats, which presents a big challenge. Do we encode as "one size fits all"? Do we encode just High, Mid and Lo quality streams? Do we encode for Progressive Download? Do we transcode on the fly? Where do we do the transcoding? At the core, edge or home?

Creating adaptive streaming services typically requires multiple encoders or transcoders per service depending on the source content format and the desired client device formats and bit rates. These encoders must be tightly synchronized to produce a valid adaptive streaming bouquet where each service instance starts and ends at the same point in time, and with the proper bit stream semantics such that client decoders can seamlessly switch between streams within the bouquet in a seamless manner.

Transcoders that are able to deliver high quality at lower bit rates can be a significant investment, especially since several are needed to produce the appropriate adaptive streaming bouquets for the three screens. Each of these then has a preferred resolution or encoding profile in terms of the device capabilities as well as subscriber expectations in terms of delivered picture quality. This cost favors centralizing the adaptive streaming transcoders into one or two super head ends for larger operators or possibly in a hosted service offering for smaller operators. Transcoding or re-encoding closer to the source also allows the provider to better control the video quality. Offsetting this, however, is that such centralization requires more backbone bandwidth to distribute these bouquets of services around a provider's footprint, and as discussed later, it impacts storage costs for the servers and CDN.

An alternate concept that has been proposed for unicast streams is to use low cost dynamic transcoders at the edge of the network that respond directly to the client's bandwidth requests in real-time. A key advantage of this approach is that such a transcoder could deliver exactly the requested bit-rate to fit the available bandwidth, offering a wide variety of bit rates. This compares to an adaptive bit-rate bouquet that was prepared farther back in the network might only have three or four discrete bit-rates to choose from at each chunk boundary. Offsetting this, however, is that low cost transcoders require higher bit-rates over the IP access network to deliver the same quality as a higher quality transcoder. And, since the number of unicast sessions can be very dynamic, the head end would need to be provisioned with enough edge transcoders to meet the recurring maximum loads, but would likely be underutilized most of the time.

Ad Insertion

A significant advantage of adaptive streaming is that it enables efficient ad insertion. Since the client device requests content by requesting the next appropriate chunk via a URL in a play list, the play list can be modified dynamically by either the server or client application to substitute the ad play list URLs in that appropriate locations of the media play list based on SCTE 130 signaling. This enables seamless insertion of targeted ads either in the network, or by pre-placing the relevant targeted ads onto a home gateway or DVR client and inserting them locally. In either case, the splice is entirely transparent since adaptive streaming chunk boundaries are always created to allow seamless switching from one stream to another.

By adding this ad substitution intelligence into the network servers, ads can be more dynamically targeted and overall advertising management is simpler since there is no need for a system to pre-place ads into subscriber devices. This server based approach also enables the same ad insertion capabilities across the entire range of client devices including those with storage such as DVRs and those with very little "extra" memory such as inexpensive IP set tops or smart phones.

IMPLICATIONS FOR SERVERS & DELIVERY NETWORKS

Video Delivery

The shift to Adaptive Streaming imposes significant changes to the roles of servers and delivery networks in providing video service to consumers.

As noted earlier, Cable video delivery has relied upon a push streaming model where the server is responsible for maintaining stream pacing. Network transport has typically been based on UDP. To maintain correct buffer behavior the stream is constructed to meet the requirements of the MPEG-2 Systems buffer model. Adaptive streaming however evolved from Progressive Download and is based on the client pulling segments of content over a reliable network transport as it requires them. This shifts much of the burden of pacing and buffer management from the server to the client.

Progressive Download grew out the need to deliver video over HTTP connections, and it is possible to deliver Adaptive Streaming content with simple web servers. This approach is viable for lab trials and small scale deployment. However, to successfully grow to large scale deployments may require that servers and other CDN components have some degree of media and session awareness.

Content Storage

The requirement to store multiple bit rate representations of each content essence creates additional demands for library storage. If the content is stored in 5 different bit rates over a 2:1 range (for example, an SD stream that ranges from 2 Mbps down to 1 Mbps in 0.25 Mbps increments) the storage requirement is 3.75 times that required for the highest quality stream alone. This does not reflect the overhead of any system support files, such as index or trick mode files, which may also have to exist in multiple bit rate versions. This increased storage requirement will impact library sizing and potentially edge caching and CDN bandwidth requirements.

If simple web servers are used to host the multiple bit rate versions it may be necessary to store each fragment in a unique file. For example, to store one hour of content using the above assumptions and a fragment duration of 2 seconds would require 9000 fragments. When considering a large content library, the number of files quickly becomes excessive and may require special attention to file system tuning and layout.

As mentioned earlier, it is better to use a container file format that allows fragments to be rapidly identified and accessed, such as a fragmented MP4 file, or a stored transport stream with segmentation markers. A media aware server can then respond to requests for systematically named fragments or fragments specified by Normal Play Time (NPT) range by extracting the requested segment from the container file.

Trick Mode Support

Support for VCR style trick modes (that is, scrubbing forward or backward through the content at faster than real-time) has traditionally been a feature of Cable on demand services. In a Progressive Download environment, this style of trick mode operation, when available at all, is restricted to operating on content that has already been buffered in the client, combined with the ability to restart the download, and normal play, at a client specified offset.

In order to replicate the existing Cable On Demand experience, it will be necessary to add mechanisms to provide VCR style trick modes in an adaptive streaming environment. This can be done either in the client or in the delivery network. Some systems have explored alternate ways of displaying trick mode operations that may be more suited to adaptive streaming, for example popping up a filmstrip of thumbnail key frames as a navigation aid

CDN Bandwidth & Caching

One of the attractions of adaptive streaming based on fragmented content is that it is a good fit for the use of a Content Delivery Network to efficiently provide content to the consumer. However caching algorithms designed for traditional web traffic may not result in optimal use of network bandwidth and cache storage. In extreme cases they could result in pathological overuse of resources.

Consider the example of a consumer viewing a movie during a time when available bandwidth is varying. At the end of the session the collection of fragments in the nearest edge cache represents the bandwidth available over the duration of the session. If another consumer requests the same logical content it is desirable to reuse as many of the fragments stored in the edge cache as possible. However that next session may face very different bandwidth availability and will consequently request a different set of fragments significantly reducing cache efficiency. Using a CDN architecture that is media aware would allow for more efficient use of the cache and CDN bandwidth.

Earlier we discussed the application of adaptive streaming to ad insertion by replacing or inserting chunked ad content. A media aware network could move this processing to the edge cache. In this application the media aware edge cache could function as an Ad Decision Manager in an SCTE130 ad insertion system.

Introducing media awareness into the CDN also helps it protect itself against misbehaving or malicious clients. Media aware servers can place bandwidth limits on client requests and detect access patterns that do not match the content attributes.

EFFICIENCY AND VIDEO QUALITY

Is adaptive streaming as efficient as traditional methods of video distribution, namely Variable Bit Rate (VBR) and Constant Bit Rate (CBR) video? That is the question we explore in this section.

VBR, CBR and P-CBR

VBR is widely used to deliver video because it is capable of producing constant video quality. In VBR methods, an encoder uses as many bits as necessary to achieve a constant target video quality. As a result, bit rate varies freely in time but no bits are wasted, at least in theory.

CBR is also widely used because bandwidth resources can be allocated with virtually no uncertainty. In CBR, an encoder causes video quality to fluctuate up and down so as to achieve a fixed target bit rate.

Adaptive streaming may be thought of as a special form CBR know as Piecewise-Constant Bit Rate (P-CBR) because every adaptive streaming chunk has a constant bit rate over its duration.

At first glance, it might seem that P-CBR would be like its parent, CBR, in the sense that video quality would not be constant. Perhaps surprisingly, P-CBR is just as capable as VBR of delivering constant video quality.

Data that illustrate constant-quality P-CBR is shown in Figure 2. The thin “noisy” line in Figure 2 shows an example of a constant-quality VBR stream. The thick flat line shows a P-CBR stream that would also result in constant video quality -- in fact, the same video quality as for the VBR stream. Both streams produce the same constant video quality because the total number of bits delivered during each piecewise-constant interval is the same for the P-CBR stream and the VBR stream.

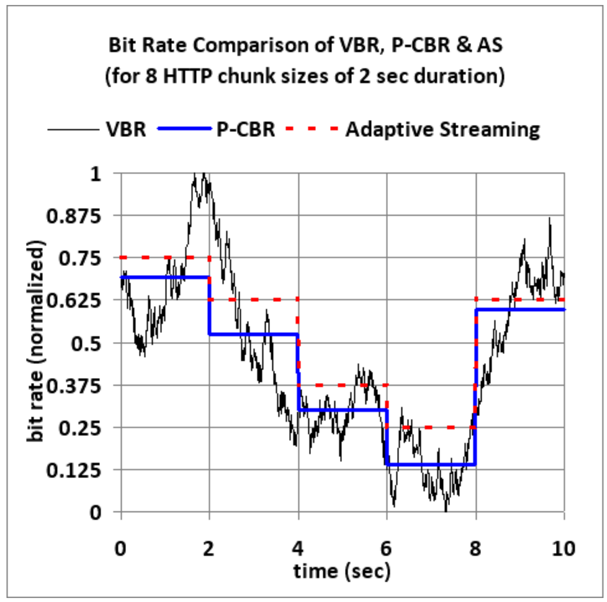

Figure 2. – Adaptive Streaming: Basics

At the end of each piecewise-constant interval, a decoder has all the bits it needs to reconstruct the preceding video. The real difference between VBR and P-CBR is latency, not video quality. P-CBR introduces intrinsic latency because the decoder needs to wait until the end of each piecewise-constant interval to be sure it has all the bits it needs.

Adaptive streaming protocols are also capable of delivering video quality that is as good as VBR, provided the client has sufficient bandwidth. This is a limiting case in which adaptive streaming may be imagined as a coarse quantization of a P-CBR stream that is equivalent to a VBR stream in all but latency. But adaptive streaming also has the flexibility to reduce bandwidth requirements if needed by switching to a lower resolution or quality version of the content. So, adaptive streaming offers service providers the ability to deliver video quality comparable to VBR, while managing bandwidth in a simple manner like CBR.

ADDITIONAL CONSIDERATIONS

Adaptive streaming protocols are not enough by themselves to enable new approaches to cable IP video delivery. Adaptive streaming introduces new challenges, such as managing the myriad chunks, media fragments, and associated metadata. Fortunately, enforcing media policies and managing assets are not entirely new problems.

Solutions already exist for enhanced asset management systems (AMS) that are designed to package, integrate, manage, and deploy content from many different sources and distribute those assets across multiple platforms. In this adaptive streaming context, asset management challenge may be viewed as an extension or evolution of existing on-demand asset management. While adaptive streaming will come with new business and technical issues, it is reassuring that some of the challenges of dealing with “infinite catalogs” have been addressed already and can serve as a foundation for future progress.

Since Adaptive Streaming changes the underlying transport of video services, this will also generate the need for new tools for service monitoring. Service providers will want the capabilities to be able to measure the Quality of Experience. For managed IP Video services, it will always be critical to maintain video quality.

Cable Migration Strategies

Many cable system lineups today are full with a wide mix of analog, digital and high definition video services in addition to video on demand offerings, high speed data service and telephony service. Meanwhile, providers are under pressure to add additional HD services and upgrade high speed data to DOCSIS 3.0, both of which require significant additional spectrum on the cable plant. Migration to IP Video will put even more pressure on bandwidth needs.

There are many tools available today that enable cable operators to recover existing or gain new spectrum capacity in their systems. These include analog reclamation by moving analog services to digital, enabled by low cost digital terminal adapters (DTAs), Switched Digital Video (SDV), migrating services from MPEG-2 to MPEG-4 video, node splits, and HFC expansion up to 1GHz.

Using some or all of these tools to free up spectrum enables cable operators to deploy additional DOCSIS bandwidth needed for IP Video. Cable operators can deploy additional DOCSIS 3.0 bonding groups and begin using these for adaptive streaming of media. This can initially be deployed to PCs and smart phones in the home via WiFi connections. If sufficient IP resources are available, the provider can also support IP set tops connected to the managed home network. These offerings can be used for both new linear TV channels or additional on demand content and also can take advantage of MPEG-4’s efficiency improvements.

During this transition period, many current set tops can be used in a hybrid configuration, using both their QAM and

DOCSIS capabilities to deliver the operator's full suite of services to their subscribers. And, over time, as more of these hybrid set tops or new hybrid gateways are deployed, cable operators can begin migrating some of their traditional QAM VOD and linear services to the IP path.

The hybrid home gateway enables the use of low cost IP-only client set tops elsewhere in the home. Eventually, when all services migrate to IP, even the gateway set top can become an IP-only device and this leads to a simpler overall system architecture that has the potential to support all IP clients across all three screens, the TV, PC, and phone, in the subscriber's home.

CONCLUSION

Adaptive streaming is an emerging technology that is of great interest to cable operators for deploying IP Video. It grew out of internet video delivery and provides the smooth user experience of streaming with the ability to scale economically like Progressive Downloads thanks to HTTP transport. Several proprietary adaptive streaming ecosystems are already in place and standardization efforts are underway. It is an obvious choice for providing services to the 2nd and 3rd screens (i.e. PC and mobile devices).

After the adaptive streaming overview, the paper took a look at using adaptive streaming in cable environment. It holds many promises and challenges. Some of the benefits include: bandwidth efficiency; minimal local storage required in user devices; support for trick mode; simplified synchronization between server and client; and expanded opportunities for targeted advertising.

We took a closer look at the impacts on the video servers and distribution networks. Some of the challenges that need to be addressed are the pressure on increased content storage and CDN bandwidth. Trick mode support and caching algorithms are other important areas that are impacted.

Finally, the paper took a deeper dive into the bandwidth efficiencies and video quality of Adaptive Streaming compared to today's VBR and CBR delivery. Adaptive Streaming holds the promise of video quality comparable to VBR with the ease of bandwidth management like CBR.

Adaptive Streaming will create new high quality multi-media distribution opportunities. It will enable rich user experiences as well as monetization of video delivery. This technology will become a cornerstone of future cable IP Video delivery systems.

Contact Info:

For more information, contact John Ulm at julm@motorola.com .

References:

1. Move Networks, http://www.pr-inside.com/move-networks-announces-microsoft-as-r773575.htm

2. Move Networks, http://www.movenetworks.com/history.html

3. Apple adaptive streaming IETF submission: http://tools.ietf.org/html/draft-pantos-http-live-streaming-01

4. Introduction to Data Compression. Khalid Sayood. Morgan Kaufmann Publishers, San Francisco, CA 2006

SIDEBAR – SOME BACKUP MATH

Efficiency of Adaptive Streaming

It is useful to investigate a limiting case to understand the bandwidth efficiency of adaptive streaming. Consider a scenario in which a provider wishes to deliver video that meets or exceeds a particular operational video-quality level. If the video is delivered using VBR, we could produce a stream such as the one represented by the thin line shown in Figure 2. If instead a P-CBR method is used with regularly spaced piecewise-constant intervals, it would produce a stream such as the one represented by the thick line in Figure 2. However, for adaptive streaming the chunks would be able to take on only certain pre-defined bit-rate values, and we would produce a stream such as that represented by the dashed line in Figure 2.

Recall that the number of bits delivered by the P-CBR stream in each interval is the minimum number of bits needed to achieve the target video quality. Thus, in the limiting case, the bit rate associated with each adaptive-streaming chunk must be chosen so that the total number of bits delivered during each interval is equal to or greater than the total number of bits delivered by the P-CBR stream during the same interval. More often than not, the adaptive-streaming chunks will end up delivering more bits than the minimum necessary. Those extra bits are the overhead associated with adaptive streaming.

Bit Rate "Overhead" and Adaptive Streaming

The amount of overhead, or excess bit rate, associated with adaptive streaming depends on the number of quantization steps: i.e. the number of possible chunk sizes. More chunk sizes translate into finer precision and less mismatch between the P-CBR bit rates and the adaptive streaming bit rates.

The mismatch is a quantization error which can be analyzed according to standard methods such as those described by [4]. If we make the least presumptive assumption that the value of the quantization error has a uniform probability distribution, then the average quantization error, B_{mqe}, would be equal to one-half of the difference between chunk sizes, ΔB. Thus we may write $B_{mqe} = \Delta B/2$.

Given a maximum operational VBR bit rate of $B_{\max}$, if our adaptive streaming protocol employs a number of uniformly distributed chunk sizes represented by N_{chunks}, then we may write:

$$B_{mqe} = \frac{1}{2}\frac{B_{\max}}{N_{chunks}} = \frac{B_{avg}}{N_{chunks}}$$

Note that we use our assumption of uniform probability to substitute $B_{avg} = B_{\max}/2$ in the above equation, but the choice of probability distributions does not affect our conclusions in a meaningful way for the purposes of this paper.

The average quantization error, B_{mqe}, is the overhead of the limiting case of adaptive streaming. It is the extra bits that would need to be delivered to a client to match or exceed the video quality delivered by a VBR stream. Note that the average quantization error is inversely proportional to the number of chunk sizes.

In the simplest view, if we were to use 10 chunk sizes, we would expect an overhead near 10%. Five chunk sizes would correspond to an expected overhead of 20%. Eight chunk sizes would be equivalent to 12.5%.

Coding Precision and Adaptive Streaming

The simplest view is not, however, the complete view. The overhead in adaptive streaming represents real data that goes towards improving video quality above that of the corresponding theoretical VBR stream. The more sophisticated view of adaptive streaming is that is not only a form of piecewise-constant bit rate, it is also a form of piecewise-constant video quality. Each piecewise-constant interval of adaptive streaming delivers a level of quality that could be matched by a VBR stream having the same average bit rate over the interval..

For H.264/AVC, video quality is largely regulated by a coding-precision parameter known as QP, which takes on positive integer values up to 51 with lower values corresponding to higher video quality. The H.264/AVC standard is designed so that a change in the QP value by 1 will tend to produce an average bit rate change of approximately 12.5% regardless of absolute bit rate.

Thus, in the example of 8 chunk sizes, adaptive streaming would produce an average video quality boost approximately equivalent to a unit change in the average QP value. The general form of the relationship between the overhead associated with adaptive streaming and the increase in effective coding precision ΔQP_{eff} (video quality) may be written as shown below:

$$\Delta QP_{eff} = -\frac{\log\left(1 + N_{chunks}^{-1}\right)}{\log(1.125)}$$

What the above equation indicates is that an adaptive streaming application that employs 5 chunk sizes, for example, and produces exactly the same average bit rate as a VBR application would result in a loss of coding precision of approximately 1.5 QP values. Ten chunk sizes would alter the effective QP value by approximately 0.8 on average, which would not normally be noticed by a typical consumer.