WILL HTTP ADAPTIVE STREAMING BECOME THE DOMINANT MODE OF VIDEO DELIVERY IN CABLE NETWORKS?

Michael Adams
Ericsson Solution Area TV

Abstract

There is a great deal of interest in HTTP adaptive streaming because it can greatly improve the user experience for video delivery over unmanaged networks. Adaptive streaming works by adapting, in real-time, to the actual network throughput to a given endpoint, without the need for "re-buffering". So, if the network throughput suddenly drops, the picture may degrade but the end-user still sees a picture.

Although adaptive streaming was originally developed for "over-the-top" video, it brings significant advantages in managed networks applications. For example, operators could set session management polices to permit a predefined level of network over-subscription rather than blocking all new sessions. This flexibility will become more and more important as subscribers start to demand higher quality feeds (1080p and above) and 3D programming. Meanwhile, adaptive streaming increases transport overhead, requires multiple bit-rate encoding, additional buffering and synchronization, and two-way network connectivity.

Not very long ago, Internet Protocol (IP) was seen as a niche protocol best used for delivering datagrams over unreliable networks. Today, IP has become a ubiquitous transport protocol for every application over every possible physical layer. This transition happened rapidly despite the additional overhead and complexity of IP compared with protocols like SONET and ATM. Will the same become true for adaptive streaming protocols? Will they quickly dominate, as every new consumer electronics device ships with support for adaptive streaming? Will the ubiquitous nature of adaptive streaming trump any loss of efficiency that it brings?

This paper will describe what makes adaptive streaming different from other modes of video delivery, and how adaptive streaming works. It will discuss the pros and cons for adaptive streaming and analyze to what extent it will become the dominant mode of video delivery in cable networks.

INTRODUCTION

HTTP adaptive streaming is the generic term for various methods of adaptive bit-rate streaming over HTTP. These include:

- Adobe Dynamic Streaming for FLASH [1]
- Apple HTTP Live Streaming (HLS) [2]
- Microsoft Smooth Streaming for Silverlight [3]

Although each the above are different implementations of adaptive streaming, they have a set of common properties:

- Content is encoded at multiple bit-rates
- A point-to-point HTTP connection is used to deliver the content stream from a server to a client
- The bit-rate can be changed on the fly to adapt to changes in available network bandwidth
- The client is responsible for fetching multimedia data (‘client-pull’) from the network.

Standardization

MPEG Dynamic Adaptive Streaming over HTTP (DASH) is set to reach Final Draft

International Standard status in July, 2011 [4]. It is based on the 3GPP Adaptive HTTP Streaming specification.

Meanwhile HTTP Live Streaming has been submitted as a draft informational proposal to the IETF [1].

Nevertheless, it is unclear how long it will be before a common approach is widely adopted. The most likely prospect is that multiple adaptive streaming implementations will continue to co-exist and that cable operators will choose to support the most popular variants in order to support a broad range of consumer devices appearing in the marketplace. Apple appears to be an early winner in the tablet space where the iPad supports only HLS. Meanwhile Silverlight and Adobe FLASH are both being used for streaming to PCs.

A recent development at the time of writing (April 18, 2011) is that Adobe has announced support for Apple HLS in their latest Flash Media Server.

Advantages

Adaptive streaming brings a number of key advantages to the network operator when compared to its close cousin, progressive download. These include:

- The ability to change video bit-rate on the fly, allowing the client to select the best stream according to network throughput, which can be indirectly measured by monitoring the receive buffer.
- Only content that is actually watched traverses the network.
- Secure DRM based on content encryption rather than secure HTTP.
- A seamless mechanism for real-time ad insertion.
- Fast channel change implemented by selecting low bit-rate stream first.

Although these properties are important, this paper will focus on how adaptive streaming compares with MPEG-2 transport, which is the dominant mode today for delivering content to the cable set-top box.

HOW DOES ADAPTIVE STREAMING WORK?

The easiest way to understand how adaptive streaming works is to start with its close cousin, progressive download.

Progressive Download

Progressive download incorporates two functions that are coupled together:

1) Download - an HTTP session is established to transfer the content file from the server to the client device.

2) Playback - once the client estimates that the receive buffer is sufficiently full, it starts to play the file from the head. If the network bandwidth is constant, the playback will continue uninterrupted because playback of the file will always lag download of it.

A major flaw in progressive download is that if the playback rate exceeds the download rate eventually the buffer will be exhausted and playback will freeze. This ‘re-buffering’ is extremely frustrating to the user.

Another flaw is that progressive download does not include any provision for flow control. In conventional (for example, MPEG-2 transport) delivery of video and audio packets, the rate of transmission is synchronized to the bit-rate of the payload. This is an important function that is necessary to prevent buffer underflow or overflow at the receiver. In contrast, progressive download treats the payload as just another file that should be downloaded as fast as the network permits. Thus progressive download may consume network resources in a burst of traffic until the file transfer is completed.

In practice, some progressive download servers implement a throttling mechanism to cap the maximum download rate to slightly more that the payload bit-rate to prevent this kind of behavior.

Adaptive Streaming

Adaptive streaming makes changes at the server and the client to increase the overall quality of experience of the end-user (viewer). These changes also directly impact the network characteristics of adaptive streaming. Finally, these changes pave the way for extensions to enable the delivery of live streams.

1) Multiple bit-rate encoding

To support adaptive streaming, the content must first be encoded at multiple bit-rates which must be pre-defined by the operator to provide an acceptable tradeoff between quality and bit-rate. In order to reduce the bit-rate, the content can be encoded at lower resolution and/or lower frame rates than the source.

2) Segmentation

Adaptive streaming sub-divides the encoded multimedia content into segments (or “chunks”). The segments are typically quite large containing between 2 and 10 seconds of multimedia content. Each segment can be delivered at a different bit-rate because it is aligned to the Group of Pictures (GOP) structure.

When combined with a set of encoding sessions of the same content, at different bit-rates and qualities, segmentation allows a client to switch from one stream to another seamlessly. Each stream is called a profile in streaming parlance.

In order for the client to be able to select segments from the appropriate profile for the stream, a manifest file is created. This is essentially a set of pointers, within a media file or list of media files, allowing the client to access the next segment at the desired bit-rate.

3) Adapting to Network Throughput

The adaptive part of adaptive streaming is enabled at the client rather than the server. The client continually monitors the available bandwidth and the media being delivered, and will dynamically switch to a higher or lower bit-rate session in order to keep the receive buffer within set limits. Seamless adaptive streaming means that the user sees no visible interruption in this process, because segments are aligned to closed GOP boundaries.

4) Flow control

Flow control is almost a misnomer as it is an indirect effect of segmentation and client behavior and not an explicit goal of adaptive streaming. However, the result is similar:

- The client is designed to download a sufficient number of segments in order to prevent buffer starvation in the event of network congestion. Clients may use different algorithms, but approximately 30 seconds of buffering is common.
- In the steady state, once the buffer is sufficiently full, the client will only request the next segment when a segment is played, essentially synchronizing the fetch rate to the play out rate.
- If network congestion occurs, the buffer will begin to empty, but the client will start to request segments encoded at a lower bit-rate to compensate.
- In the new steady state (if congestion persists), the client will once again synchronize to the (lower) play out rate.

It is important to note that all of the above assumes a given client behavior. Certain clients may be extremely conservative and attempt to maintain a much larger buffer (in fact, it would even be possible to modify a client to emulate a progressive download by continuing to fetch segments regardless of buffer fullness).

5) Live Content

The addition of segmentation means that adaptive streaming can be used to deliver live content in real-time. This is achieved as follows:

At the encoder:

- A set of real-time encoders is used to encode the source content at multiple bit-rates.
- As before, the output of each encoder is segmented according to the GOP structure of the video. Segments are buffered in memory (for a limited period of time).
- The manifest is updated in real-time, providing the client with an index of segments.

At the client:

- After the channel has been selected, the client will download the manifest file, and then starts fetching segments. The first segments are actually a few seconds behind the current time because a certain receive buffer size (typically in the order of 10 seconds) is needed to maintain a smooth play out.
- If network conditions are good, the client will rapidly fill its receive buffer but continue to play the content with significant delay.
- Once a steady state is achieved, the client will continue to fetch packets as soon as they are published by the server. In other words, it will frequently re-download and re-check the manifest file to see if new segments are available for download.
- In the case of network congestion, the client will fetch smaller segments (that is segments encoded at lower bit-rates) to ensure that the receive buffer doesn't underflow.
- In the case of severe network congestion, the client will have to pause play out and restart the process.

As can be understood from this description, "live" is a term that is loosely applied in this context. While it is true that there is a delay in the play out of MPEG-2 transport streams at the client, this is constant and tightly controlled by the specification. In contrast, the behavior in an adaptive streaming client is poorly specified and delays introduced are much larger.

APPLICATIONS IN CABLE NETWORKS

Adaptive streaming has different applications, each with different implications for cable operators. Beyond "over-the-top" (OTT), the most obvious fit is to supplement or replace on-demand services, and this leads to a fairly straightforward comparison with existing techniques. The second is live streaming for "broadcast" programming which has a much larger potential impact upon the network.

On-demand Services

On-demand services are implemented in cable systems to minimize the impact on the set-top box. In the first commercial deployments of on-demand (circa 1998), the most expensive part of the system was the set-top box, which was optimized for playing a standard MPEG-2 multiple program transport stream carrying MPEG-2 video and AC3 audio, the ATSC standard for digital broadcasts. Therefore on-demand systems were specified to emulate a broadcast stream as closely as possible. Some minor changes were made:

- A constant bit-rate format SPTS is specified. Initially at 3.75 Mbps for standard definition video, and later at 15 Mbps for high definition video.
- Initially, conditional access encryption was ignored and subsequently a fixed-key scheme was used (in contrast to broadcast streams where keys are updated continuously).

None of these changes made the set-top box more expensive. All the differences in operation were software changes related to the program guide and signaling.

Taking an existing on-demand system and extending it to provide the same service using adaptive streaming can be done by adding components to the existing system as follows:

- A content management system to manage encoding of assets into multiple bit-rate MPEG-4 AVC format.
- An offline encoding system that is aware of GOP boundaries and is able to create synchronized segments of video/audio payload.
- A server capable of servicing HTTP requests from the clients and delivering the multimedia payload in the chosen adaptive streaming format. The server must also publish a manifest that indexes each segment at each chosen bit-rate.

The result is that new clients that support adaptive streaming can now access the same on-demand library offered to the set-top box.

In practice, a Content Delivery Network (CDN) will be used to scale the system. This takes advantage of a property of HTTP that it can be cached. Thus a subsequent request for the same stream by a different client could be serviced by the CDN transparently.

Broadcast Services

Providing broadcast (that is “live”) services to an adaptive streaming client is quite a different challenge for cable operators.

A broadcast channel must encoded in real-time into the adaptive streaming format since content is not available ahead of time.

Taking an existing broadcast system and extending it to provide the same service using adaptive streaming can be done by adding components as follows:

- A real-time encoding system that is aware of GOP boundaries and is able to create a synchronized encoded payload
- A system that segments the video/audio payload
- A server capable of servicing HTTP requests from the clients and delivering the multimedia payload in the chosen adaptive streaming format. The server must also update the manifest file in real-time.

In the live streaming case, a CDN that is optimized for streaming media is essential since the encoding system will be located in the core of the network.

SERVICE CHARACTERISTICS

With the introduction of adaptive streaming, the cable operator is moving into a new realm of operation with many more aspects of service delivery being now out of their control. For example, if we compare the buffer model of a set-top box, it is well defined and implemented according to MPEG-2 transport systems. Extensive testing and analysis of large-scale systems has been done to ensure unexpected side effects do not affect the delivery of video to the device. In contrast, the cable operator has little control over the algorithm implemented by a particular iPad or PC connected to their network. If one client implementation is not well behaved could it negatively affect performance of other well-behaved devices on the same network segment?

What are the potential impacts on the service, as perceived by the subscriber, as adaptive streaming is introduced into their viewing experience?

Delay

In most implementations of live streaming, the experience is still significantly more delayed than with current MPEG-2 transport delivered services. One study found that adaptive streaming introduced an 8 second delay [5]. This compares unfavorably with delays due to encoding/statistical multiplexing which are usually about 2 seconds at worst.

This means in practice that a subscriber may notice that the video on their iPad (for example) is significantly delayed compared to the video on their TV (while watching the same programming on both devices in the same room). In practice there is no technical solution to this as each client device may implement a different algorithm and attempt to maintain different playback buffer sizes, and therefore introduce differing playback delays.

Channel change

Adaptive streaming clients are typically designed to start streaming at the lowest acceptable bit-rate after a channel change and then rapidly increase the bit-rate selected according to network throughput. This provides a useful fast channel change mechanism.

Stability

If multiple adaptive streaming clients contend for limited network bandwidth, as one client reacts to congestion it has the effect of making more bandwidth available to the other clients. In certain circumstances, a feedback loop can be created leading to instability.

The effect of this is that the client may constantly switch between different bit-rates generating an annoying artifact, visible to the user as a repetitive cycle of softening and sharpening of the picture.

One solution is to avoid over-subscription, and therefore congestion, of the network. However, this means giving up a potential benefit of adaptive streaming, namely the ability to allow over-subscription during peak demand.

Customer support and Troubleshooting

Taking a scenario where the client fails to deliver a smooth, acceptable quality video stream, how can the trouble call be resolved? This represents one of the biggest challenges that will be faced by operators as adaptive streaming is widely deployed:

- Is the problem in the network or the client?
- If it is a network congestion issue, what kind of real-time trace can be performed to identify the source of congestion?

Ad Insertion

Client-side ad insertion has become the dominant model used with adaptive streaming. In this case the player makes a local decision to insert an ad before a requested video, or between videos clips in a play-list. The reason for this approach is targeting – the ad can be targeted to the individual subscriber based on known preferences or based on recent searches.

Adaptive streaming of broadcast content will, in many cases, include traditional ad spots. It would be possible to mark ad avails using SCTE 35 data in the manifest file, allowing ad insertion to be accomplished at the client (to replace the network ads). It is technically challenging to implement network ad insertion, because the CDN would have to be aware of SCTE 35 information and perform re-direction based on geographic or other parameters.

NETWORK CHARACTERISTICS

As adaptive streaming becomes a more significant source of video content how will this affect overall network utilization? What are the likely impacts to the cost of video delivery over IP networks compared to the traditional deliver using MPEG-2 transport streams over QAM?

Assumptions

On-demand content will continue to be delivered as a constant bit-rate MPEG-2 Single Program Transport Stream (SPTS) to existing devices such as deployed set-top boxes. The standard rates for these streams are 3.75 Mbps (Standard Definition) and 15 Mbps (High Definition).

As new devices, such as smart TVs, iPads, Tablets, etc. appear in homes, cable operators (notably Comcast, TWC, and Cablevision) are starting to support adaptive streaming to these devices. This trend will continue and more and more programming will be delivered using adaptive streaming.

In many cases the final connection to the device will be by WiFi, and is therefore subject to instantaneous fluctuations in throughput due to changes in propagation, including distance from the WiFi router, and other devices operating in the same limited frequency band.

Traffic Profile

Adaptive streaming has an interesting, and potentially very useful, traffic profile that makes it attractive to the cable operator:

- It plays well with other TCP-IP traffic since all TCP-IP traffic reacts to congestion is a predictable way.
- It takes advantage of the maximum network throughput available up to a limit, which is set by the highest bit-rate encoded version of the content.
- It automatically responds to network congestion by progressively reducing the bit-rate of the content (according to pre-defined bit-rates determined by the operator).

If adaptive streaming and general Internet traffic are distributed over the same network infrastructure, using differential quality of service mechanisms to tag video traffic as higher priority is recommended.

Overhead

How much overhead does HTTP adaptive streaming add when compared to MPEG-2 transport stream? The increased overhead in the forward direction is mainly due to the additional headers for HTTP and TCP/IP. Meanwhile, in the return direction, TCP-IP acknowledgements introduce an entirely new traffic flow.

However, adaptive streaming uses the latest, most-efficient compression algorithm (most likely, but not always, H.264, otherwise known as MPEG-4 AVC). In comparison to MPEG-2 compression the bit-rate is significantly reduced, approximately halved in fact.

Fairness

There is no guarantee that different streams over the same network segment will receive equal shares of the available bandwidth. It is quite possible that one device may hog the bandwidth (due to a more aggressive algorithm) while another may be starved (due to a more conservative algorithm). This behavior could cause disruption to other services.

Burstiness

Adaptive streaming is fundamentally different from MPEG-2 transport in that, for the duration of the download of a fragment, the HTTP transfer will consume as much network bandwidth as is available, generating a bursty traffic profile.

This behavior can be modified by limiting the maximum rate of each segment download to slightly greater than the maximum bit-rate. This technique requires more intelligence in the content distribution network (CDN).

Comparison with MPEG-2 Transport Streams

The current dominant mode of video and audio delivery in cable systems is based on MPEG-2 transport streams. How do MPEG-2 Transport Streams compare with adaptive streaming?

1) Broadcast

Statistical multiplexing is universally employed to pack more channels into each QAM channel on the network. Since zero packet loss can be tolerated, the multiplexer must analyze the video payload in real-time and, during peaks of traffic, reduce it by re-quantization of DCT coefficients. This makes statistical multiplexing a relatively expensive process that can only be justified for broadcast streams.

In comparison, adaptive streaming achieves a similar result by dynamically selecting the streaming profile according to traffic conditions. However, adaptive streaming does this by using a unicast delivery model. This means that each unique viewer of a given broadcast generates a dedicated stream in the access network.

On the other hand, when a broadcast channel is not being viewed (or recorded) by any subscribers within a service group, the bandwidth allocated to it is entirely wasted. In this case, an adaptive streaming approach would consume no network bandwidth for that service group.

For the above reasons, adaptive streaming is a very poor substitute for delivering *popular* broadcast services. Introducing adaptive streaming will cause a dramatic increase in the access network traffic because of the multiplicative effect of delivering a single broadcast channel as individual unicast streams to each client.

The underlying question is whether broadcast channels in the HFC network will continue to be an efficient delivery mechanism. As subscribers move to an on-demand mode of consumption, even of news and sports programming, will any channels remain that attract enough concurrent viewers (within a service group) to justify dedicating fixed bandwidth to them?

2) Switched Digital Video

Switched digital video (SDV) is a multicast delivery mechanism, and a channel consumes no network bandwidth when it is not being watched (or recorded).

Adaptive streaming is a poor substitute for switched digital video as long as the service is *popular* (as explained above in the broadcast case). However, niche programming, that is currently delivered using SDV, would be an excellent candidate for conversion to adaptive streaming.

3) On-demand

On-demand systems have been engineered to support MPEG-2 transport streams requirement for constant delay and zero packet loss:

- Bandwidth is reserved for the duration of the session (and that bandwidth is wasted if the session is paused)
- There is a hard limit to the number of sessions
- The number of QAM channels allocated to on-demand must be over-provisioned to minimize the probability of blocking
- In the normal case, QAM utilization is relatively poor.

Adaptive streaming is therefore a good candidate to completely replace on-demand services in cable systems over time. To provide the same quality of service as today's on-demand care would have to be taken to ensure that the network is not over-subscribed. Alternatively, differential QoS could be implemented to provide different service guarantees to ensure that pay and premium content is not degraded during periods of peak demand.

RECOMMENDATIONS FOR CABLE OPERATORS

Clearly operators have little choice when it comes to supporting new devices, like the iPad, in their networks. However, should a cable operator pro-actively consider a migration to adaptive streaming for devices within their control?

As newer set-tops are specified, inclusion of adaptive streaming could bring significant benefits in terms of network efficiency:

- On-demand QAMs could eventually be retired, liberating more RF channels for DOCSIS.
- Statistical multiplexing efficiencies could be gained by sharing the pipe with other traffic types.

CONCLUSIONS

Adaptive streaming was developed to provide the best user experience for streaming of content over an unmanaged network, like the Internet. As described in this paper, adaptive streaming cannot provide a service delivery quality that matches that of MPEG-2 transport systems. In particular, adaptive streaming compares unfavorably when it comes to delay, stability, and quality guarantees. In addition, because it is a purely unicast delivery mechanism, where the client pulls content from the network, no shared bandwidth efficiencies are gained from broadcast services.

Nevertheless, adaptive streaming brings with it a level of flexibility precisely because it was designed for an unmanaged network. It allows the operator to move away from the connection-oriented bandwidth reservation system required for MPEG-2 transport systems, and, eventually, to supporting a single IP network infrastructure for all services. This approach also allows new services to be deployed extremely rapidly, a well-proven result of network transparency from the Internet model.

Adaptive streaming is here to stay because of the appearance of popular client devices – tablets, smart phones and PCs – that support only adaptive streaming. Given this reality, cable operators are already moving rapidly to add adaptive streaming capabilities to their content delivery infrastructure.

Existing set-top boxes in the network will continue to function side-by-side until they eventually become obsolete. Newer set-tops will inevitably be designed to accept adaptive streaming formats as they become standardized. Eventually, an optimized future version of adaptive streaming will become the dominant mode of video delivery in cable networks.

REFERENCES

1. Dynamic streaming in Flash Media Server 3.5 – Part 1: Overview of the new capabilities, David Hassoun, Aug 16, 2010: http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt1.html
2. HTTP Live Streaming, IETF Informational draft version 6, R. Pantos and W. May, Mach 31, 2011: http://tools.ietf.org/html/draft-pantos-http-live-streaming-06
3. Smooth Streaming Technical Overview, Alex Zambelli, March 31, 2009: http://learn.iis.net/page.aspx/626/smooth-streaming-technical-overview/
4. MPEG Dynamic Adaptive Streaming over HTTP (DASH) http://www.slideshare.net/christian.timmerer/http-streaming-of-mpeg-media
5. An Experimental Evaluation of Rate-Adaption Algorithms in Adaptive Streaming over HTTP, Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis, MMSys' 11, San Jose, CA.