Improving the Internet TV and Mobile Video Experience

An Adaptive Streaming Overview

White Paper

Internet TV and mobile video consumption continue to grow, both competing with and complementing traditional television and dedicated-network IPTV services. When delivering content over the Web or to mobile devices, the unpredictable nature of viewers' network connections – and in the case of Web-connected PCs, the playback capabilities of the viewing device – are key challenges to ensuring consistent, high-quality viewing experiences. Unlike the controlled and tightly managed network conditions of closed-network, 'walled-garden' IPTV deployments, the Internet and wireless networks offer no such assurance of consistent, sustainable bandwidth.

Adaptive streaming technologies set out to solve this problem, enabling higher-quality video with television-like continuity and reliability on the Web and mobile devices – even under drastically changing network connectivity and playback conditions. Of course, higher-quality experiences mean happier viewers who will stay engaged with the content longer, providing greater monetization opportunities for the content or service provider.

Adaptive bit rate streaming overall is not a new concept; there have been a number of implementations of this concept over the years that have met with mixed commercial or technical success. The latest generation of adaptive streaming technologies such as Microsoft IIS Smooth Streaming and Adobe's HTTP Dynamic Streaming have significantly raised the interest level and adoption of adaptive streaming while improving the resulting experience for end users. Both of these technologies use the HTTP protocol (the protocol used for Web page delivery), leveraging existing Web infrastructures and caching technologies. While these two technologies get much of the attention for Web delivery, they aren't the only technologies that are using HTTP to adaptively reach viewers – there are numerous others, as well as adaptive streaming capabilities incorporated within broader service offerings.

Much of the initial interest in the latest generation of adaptive streaming technologies focused on enabling video delivery at HD resolutions over the Web for playback on PCs. As mentioned above, however, the unpredictable bandwidth of mobile devices means they can benefit as much or more from this technology as PCs. Apple's HTTP Live Streaming for the iPhone brought adaptive streaming for mobile devices to the forefront, but adaptive bit rate delivery is also available or forthcoming on other mobile platforms from Android to Windows Phone 7. Even on PCs, the interest in adaptive streaming isn't just about HD. Adaptive streaming can also make standard definition Web experiences more consistent and reliable—of particular interest for content that isn't produced originally in HD, or in regions where limited bandwidth makes HD bit rates impractical.

Improving the Experience

At its basic level, adaptive streaming refers to a number of technologies for delivering video that adapt automatically to the viewers' network and playback conditions. As mentioned, that could be inconsistent network bandwidth, or even whether the viewer's PC can keep up with decoding and playing the video.

The latest HTTP-based adaptive streaming technologies, which we'll focus on in this paper, have roots in earlier Web video methodologies. Traditional 'basic' streaming protocols deliver live or on-demand content linearly at a single fixed bit rate. Like live television, if viewing is interrupted for any reason, it resumes from a later point in time – live content that was sent during that interruption is missed.

To accommodate variance in viewer's bit rates, we've all seen websites that offer a choice of streams at low, medium and high bit rates and resolutions. That's still common today, but not ideal. Users who aren't particularly tech-savvy may not know which to choose, and might guess wrong. Even if the user selects appropriately, he or she may be in for an unsatisfactory experience. If the High bit rate option is chosen and the bandwidth subsequently drops, the stream may begin stuttering and content may be missed. Once it starts choking up, the user may just give up watching rather than trying again with a lower bit rate. On the other hand, many viewers may choose the low bit-rate option, which will give them a consistent, reliable experience, but at a much lower quality than they could be getting when higher bandwidth is available.

For on-demand content, another common traditional methodology is progressive download. Files are downloaded from a standard HTTP Web server, with viewing of the content beginning once a sufficient percentage of the file has been downloaded. If the available bandwidth is low relative to the video bit rate, there will be frequent pauses as the download catches up. While content is not missed during network performance drops, and is simply delayed, these frequent pauses create a frustrating viewing experience. Conversely, if the available bandwidth is high relative to the video, the download may get far ahead of the viewed content. The viewer's system may download much more than is actually watched if the viewer watches only a portion of the content, wasting considerable data transfer.

Adaptive streaming avoids the user making the trade-off between quality and reliability, while also enabling faster initial playback start (rather than buffering at a specified bit rate before starting, playback can begin at a lower bit rate and ramp up) and reducing excessive data transfer. From a viewer perspective, HTTP-based adaptive streaming behaves a lot like traditional streaming protocols such as RTSP, but is really based on a series of short HTTP progressive downloads of small media segments. Even in the case of live content, the so-called 'streams' are really a series of small file downloads.

Under the Hood

At a high level, the path content takes for HTTP adaptive streaming is the same as the path it takes for non-adaptive delivery. Source video feeds or files go into an encoder or transcoder, which transforms the content into the required output formats. The outputs are sent to a server or CDN, from which they are delivered to the viewing audience. This simplified view ignores valuable aspects such as DRM and ad insertion, but represents a minimal flow.

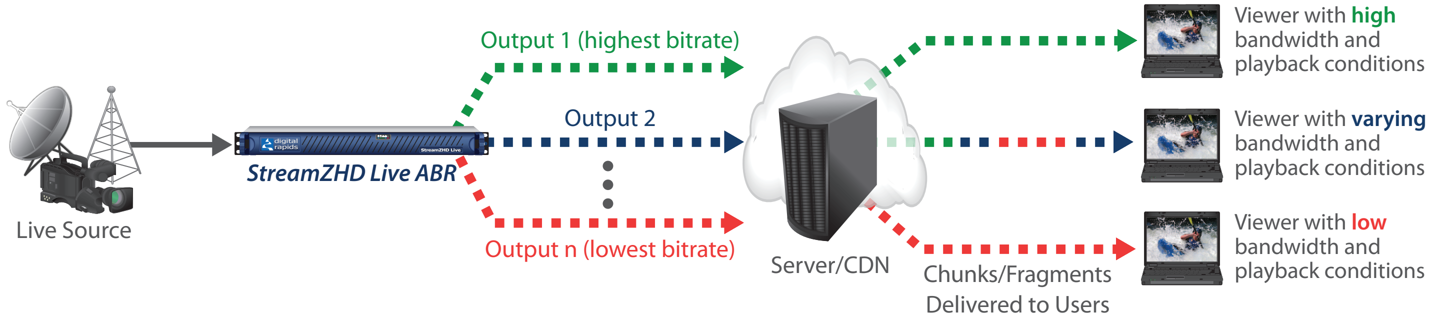


Figure 1 - Live Adaptive Streaming Workflow Overview

With HTTP adaptive streaming, the encoder or multiple encoders create several outputs at different bit rates and frame sizes from the same source. These outputs are divided up into a series of short, precisely aligned pieces of up to a few seconds in length. Depending on the particular adaptive streaming technology being deployed, these pieces might be called 'chunks', 'fragments', or 'segments', and the segmenting might be done directly within the encoder or with external tools. The segmenting occurs only at GOP (Group of Pictures) boundaries within the outputs, enabling each segment to be decoded independently of the others. In addition to the segmented media itself, manifest or index files are created to keep track of the various outputs. The client player learns which bit rates are available for the given piece of content through its manifest.

These segments are downloaded by the client player from the server or CDN over HTTP similar to a traditional progressive download, and played back in order. To adapt to varying conditions, detected through a variety of heuristics, the client changes which of the multiple outputs it is downloading segments from, effectively switching bit rates at the segment boundaries.

A visual example will make this clearer. ***Figure 2*** represents a piece of content that has been encoded at multiple bit rates, with the highest bit rate at the top, and the lowest bit rate at the bottom. Each thumbnail represents a segment, not an individual frame. The segments in this example are two seconds in duration, but the segment length is typically configurable and may vary between technologies. All of these segments are perfectly aligned, which we'll discuss shortly.

Figure 2 – Adaptive streaming technologies effectively 'switch' between bit rates by downloading segments of multiple outputs based on current conditions.

A viewer with consistently great network and playback conditions will be watching the highest bit rate segments throughout the content, while a viewer with low bandwidth or a very slow PC will be watching the lowest bit rate segments. For a viewer who has inconsistent conditions, the adaptive streaming technology will automatically select segments of the different outputs. In this exaggerated example, the viewer's bandwidth started out great but then suffered considerable network congestion, so the player switched to the lowest bit rate. As the network conditions improve, the player begins selecting higher bit rates again.

Encoding for Adaptive Streaming

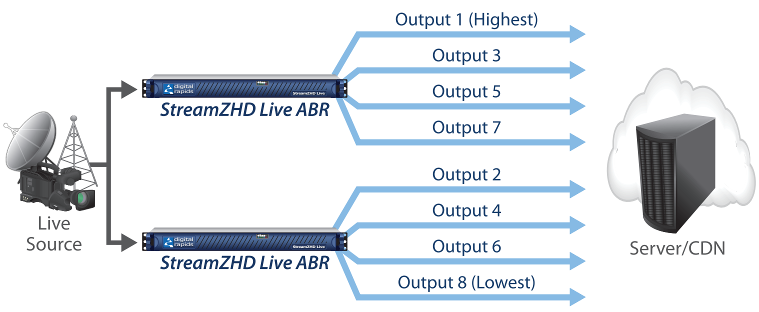
Support for adaptive streaming technologies including Adobe HTTP Dynamic Streaming, Apple iPhone/iPad adaptive streaming, Microsoft IIS Smooth Streaming and more is integrated throughout Digital Rapids' encoding and transcoding product lines, including the StreamZHD Live ABR adaptive streaming encoder, StreamZHD multi-purpose encoding system, TouchStream portable streaming appliance and Transcode Manager high-volume transcoding software. In this section, we'll look at the details of encoding for adaptive streaming, and some best practices for deploying adaptive streaming encoders.

One of the most important things from an encoding perspective is that the multiple outputs must be perfectly synchronized and aligned, from timecode down to the GOP alignment within the compressed bit streams. That's critical – if they're not precisely aligned, there can be jumps in the playback when the adaptive technology switches between bit rates. That's fairly straightforward within a single encoder, but if multiple encoders are being synchronized to increase the number of outputs or for fault tolerance, external timecode and reference sync should be used.

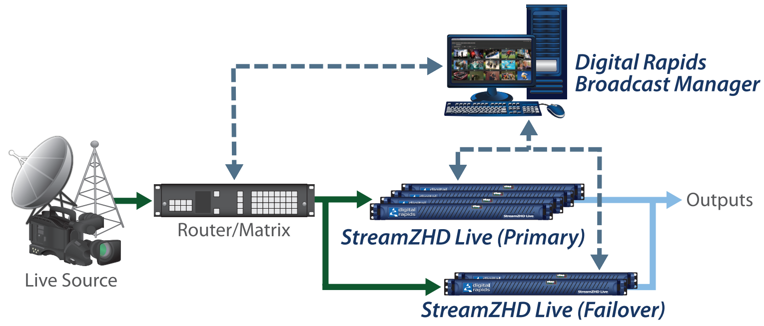
Of course, encoding for live adaptive streaming is more demanding than encoding for on-demand delivery, as all of these perfectly synchronized outputs must be created simultaneously in real time. The number of different bit rates offered for each piece of content is up to the content provider, and each technology has its own recommended practices, but live Web events to date have typically offered six or seven different bit rates.

Encoding those six or seven outputs in real time previously required multiple encoders to be synchronized and encoding the same input source. Today, those streams and more can be created concurrently in real time within a single StreamZHD Live ABR or StreamZHD unit.

Even so, users may still want to deploy multiple synchronized encoders for fault tolerance. As always, live video means no second chances, and any issues will be visible to your audience. Furthermore, while such reliability is always desirable, live HTTP-based adaptive streaming is often used for high-profile, large-audience events, making reliability even more critical. One redundancy strategy is to use two encoders for each source, with each encoder providing half of the outputs. The bit rates should be staggered between the two systems, rather than outputting all of the high bit rates from one system and low bit rates from the other. If one encoder goes down, the same content is still being streamed live, just with half the number of bit rates to switch between. And by staggering the bit rates, viewers who were watching a stream from the encoder that failed drop down by just one bit rate, not four. With the high performance of our StreamZHD Live ABR and StreamZHD systems, this often leaves unused processing power. You can take advantage of this by creating outputs for multiple platforms concurrently from each system, such as iPhone encodes simultaneously with HD for the Web.



Another strategy is to have a pool of backup units available for failover. If an encoder goes down, encoder management and automation software such as our Broadcast Manager detects this and brings one of the backup units online, while controlling upstream video routers to switch the correct input signal to that backup encoder. If there are fewer backups than primary encoders, this can reduce the total equipment cost significantly, but with the trade-off of overall redundancy and a brief interruption during failover.



When planning a live adaptive streaming deployment, it's important to consider the network connectivity between the encoders and the server or CDN. While some adaptive streaming technologies or services allow a single high bit rate stream to be sent from the encoder, with the server or CDN generating the lower bit rates and resolutions from it – at the quality expense of potential recompression artifacts – more commonly all of the outputs are sent directly from the encoder to the server. The pipe between the encoder and the server or CDN must thus be able to sustain the total of all of the bit rates being offered. Furthermore, if multiple source feeds are being encoded – such as multiple camera angles or channels – or the outputs are being sent to multiple servers or CDNs for redundancy, that total bit rate is multiplied by the number of sources and target destinations.

Similarly, when creating on-demand content for adaptive streaming delivery, it's important to consider storage requirements. There are likely to be more bit rate and resolution variants of each piece of content than previously offered with traditional streaming methods, and with the ability to now offer HD content reliably, these files may be much bigger than previous deployments.

The connectivity between the encoders and server or CDN and storage considerations are two of the factors in choosing which bit rates, and how many of them, to use for adaptive streaming. The lowest bit rate should correspond to the minimum bandwidth expected of your target audience. Selecting the highest bit rate has more considerations, including the top capabilities expected of your audience; the nature of your content (high-motion content may require higher bit rates to maintain quality); and practical considerations of both your outgoing bandwidth and delivery costs. It's also important to choose the intermediate bit rates carefully. Each adaptive technology has its own set of recommendations, but as a general rule of thumb, a variance between bit rates of 1.5x to 2x the previous bit rate is reasonable for the Web. If the bit rates are too far apart, there may be jarring changes in quality for even moderate bandwidth changes near the bit rate boundaries. If they are too close together, there may be frequent unnecessary changes with minimal quality benefit, but the increased number of distinct streams spanning the lowest to the highest will require extra outgoing bandwidth. The ideal balance provides enough bit rates to avoid dramatic changes in quality between each, but remaining practical in terms of storage or outgoing connection bandwidth and cost.

Wrapping it Up

Adaptive streaming technologies have evolved to play an essential role in bringing Internet TV a step closer to the quality and reliability of dedicated television services. Beyond the viewer benefits, content owners benefit from the resulting increased viewer engagement, while infrastructure providers such as CDNs may benefit from the scalability, capacity and cost savings available by leveraging existing HTTP infrastructures.

digital rapids®

North America
(905) 946-9666 x212
sales.na@digital-rapids.com

EMEA
+44-1428-751012
sales.eu@digital-rapids.com

Asia Pacific
+852-3972-2385
sales.ap@digital-rapids.com

Australia
+61-2-9546-1300
sales.ap@digital-rapids.com

Latin America
+54-11-3221-4202
sales.la@digital-rapids.com

www.digital-rapids.com