ADAPTIVE STREAMING
Improve Retention for Live Content
Adaptive streaming technologies make multiple video streams available to the end viewer. True adaptive bitrate dynamically switches between qualities based on the connection speed and sometimes other variables, like window size, of the viewer.

The ultimate goal of adaptive and multi-bitrate streaming is to provide an optimal viewing experience: delivering HD content to people with robust setups, while still being able to deliver content without the threat of near constant buffering to those on slower setups.

This white paper walks through an explanation of buffering, touches briefly on why it can be detrimental to viewer retention, and then delves into details on what is adaptive streaming, how to serve multi-bitrates and how this entire process is deployed at Ustream.
UNDERSTANDING PLAYBACK BUFFERING

Video streaming is achieved by sending over packets or video chunks of data. Typically, if a packet or video chunk is lost, the video player application stops receiving data until the lost packet is received. This can quickly become a problem, as later packets will continue to be delivered. For example, if packet 50 was lost, the playback of the video would stop until it's received. This occurs even as later packets, such as packets 51-75, arrive. Once 50 is finally received, packets 50-75 would be delivered all at once, causing a disruption in the video.

To avoid this problem from constantly occurring, a buffer is created by the video player. This allows the video to continue to play from the buffer even while waiting for a lost packet to be received. Theoretically, this means a video should not stop playback in order to buffer as long as the network performance and viewer's download speed is adequate.

That said, it's still possible for the viewing connection to be poor enough that the new packet doesn't arrive by the time the buffer is empty. When this occurs, the video playback has to stop and the player will wait until more data is received to continue playback. In many instances, the player keeps the video stopped for longer than is necessary to allow the player to fill up the buffer before playback resumes. Without this process, the player would quickly buffer again once the missing packet or video chunk was received.

Note: buffering can also be caused if the broadcaster does not have a sufficient upload speed compared to their bitrate or if they have issues with network stability. In this instance, the disruption can become global, affecting virtually every viewer, rather than just viewers with slower connections.

Without buffering, single packet loss can lead to losing the stream until the missing packet is resent.
BUSINESS IMPACT OF BUFFERING

For almost all use-cases, avoiding buffering should be the goal. The reason for this is that the people’s tolerance for buffering is already very small and continues to decrease. This is despite the fact that the frequency of those experiencing buffering is actually on the rise, due to the advent of higher quality and resolution content. A 2015 report by Conviva, shows that viewers experiencing buffering is actually on the rise, even though tolerance for buffering continues to diminish. This impact is measured in viewing minutes lost from the average viewer. For example, in 2011, a 1% increase in buffering resulted in an average loss of 3 minutes that the content might be viewed. By 2014, that loss has more than quadrupled to 14 minutes for each 1% increase in buffering. These figures are averages, and the study finds that the most common exit point occurs with those who experience buffering for the first time during the video content1.

The level of impact also depends on the type of content being streamed as well. For example, a 2013 study by Conviva found that the average viewing time for sports content was 40 minutes in HD, but dropped to an anemic 1 minute if the viewer encountered buffering2.

WHAT IS ADAPTIVE STREAMING?

Progressive downloads and “normal” streaming use a single encoded file as their source. Adaptive streaming on the other hand offers multiple streams for the same video content, be it live or on-demand. These streams differ, most often on the grounds of resolution and bitrate (both video and audio).

There are a variety of options available for serving content that is adaptive. These solutions range from Microsoft’s Smooth Streaming to Apple’s HTTP Live Streaming (HLS). The HLS approach functions by breaking down the stream into smaller MPEG2-TS files, also called video chunks. These video chunks vary by bitrate and often resolution. Ustream uses the HLS approach for mobile while using a proprietary delivery method on desktop to allow for HTML5 playback through using MP4 chunks.

When utilized, adaptive streaming offers a method to present a higher quality version of the feed to someone with a better setup that can support it while still being able to service those on lower bandwidth speeds through a lower quality version of the content.

METHODS TO DELIVER MULTIPLE STREAMS TO THE USER

Uploading one stream to cloud transcoding requires significantly less bandwidth at the source.

OPTION 1: Generate multiple quality streams from the source & upload them to the cloud.

OPTION 2: Generate 1 high quality stream and allow cloud transcoding to create multiple quality streams.
ADAPTIVE STREAMING: HOW A SYSTEM CHANGES BITRATES

When using adaptive streaming, the video player should shift toward an ideal resolution and bitrate setting from those available. How this process is done depends on several factors which can include:

**VIEWER’S CONNECTION**

This is the primary methodology in most setups. The system can effectively adjust the bitrate to match the best result to the download connection attempting to view the content. Most systems are conservative with this estimate, giving them at times a lower quality feed than the viewer can sometimes support. Given the impact on viewership that buffering can cause, though, this is a rational approach.

**PLAYBACK WINDOW SIZE**

A system can also monitor playback size in order to determine what bitrate and resolution combo the viewer is sent. As an example, if video content is being played back in a window that is 676x380, they would not be served a 480p or higher resolution version of the video. They would get a 360p version of that video as opposed to a 480p option, assuming both are available. The assumption is the viewer will not notice the quality difference, basically wasting the extra bandwidth needed to do this.

**AVAILABLE CPU**

Systems are also able to monitor for dropped frames. This can be used as an indicator of the CPU power available for the video decoding process. Higher quality streams, with higher bitrates and higher resolutions, require more CPU to decode.

Ustream utilizes a combination of the viewer’s connection and the playback window size in determining what bitrate to serve them initially.
Each setup looks at the resolution and bandwidth detection when the viewer first engages the video player. More robust setups will then adjust the quality based on a change in circumstances or settings. For example, the quality may be increased if the viewer’s network connection strengthens or lowered if the playback window size is reduced.

Some systems will also monitor the buffer itself in order to make this determination. For example, if a buffer stays empty for a certain duration, the system would likely make the determination that the connection is not strong enough to handle the current bitrate and make adjustments accordingly.

Ustream offers both the ability to change bitrates mid-stream and also checks the buffer to make this determination. While the first video chunk is loaded, reporting begins in the background with each downloaded chunk. The system projects available bandwidth from the viewer while they watch the stream, generating a report of their performance downloading the video chunks.

Each new report restarts the decision-making process, and the bitrate may or may not switch based on the latest report. However, if the viewer manually changes the bitrate, the player will not change their selection. For manual bitrate selection, these options are generally presented to the viewer in terms of available resolutions. However, a bitrate figure will be presented when the broadcaster is using their own encoder, rather than cloud transcoding, to deliver these bitrates and multiple streams have the same resolution.

Afterwards, the process stops for 10 minutes. After those 10 minutes, the process begins anew to determine if the viewer’s download speed has changed.

**IMMEDIATE FEEDBACK**

allows for larger chunk download resulting in a higher quality stream
CONDITIONS THAT TRIGGER THE BITRATE CHANGE AT USTREAM

Incidents that can trigger a change in bitrate in a Ustream broadcast include available bandwidth (download speed), playback window size and the performance of the buffer.

As mentioned earlier, the evaluation of available bandwidth is done through the downloading of the video chunk to determine connection strength. If the information returned indicates the download speed is able to handle a higher quality feed, this is served at the next keyframe. This determination requires that the speed is tested at a rate that is a certain percentage higher than the bitrate of the video content.

If the playback window size exceeds 20% of the current height resolution, it will change to a higher resolution, even if this might greatly exceed the current playback window size. For example, if a viewer is watching a 480p version of a video and changes their window size to 577 pixels in height (20.25% larger) the video player would give them a higher resolution, which may be a 720p version if available. This detection is precise, meaning if the playback window size was exactly 20% (576 pixels rather than 577 pixels), it would not make the change. The change in window size can take a few seconds to register, and the change in quality will occur at the next keyframe.

Note that, both of these conditions have to be met to trigger a change. For example, if the playback window size is increased to 25% larger than the video resolution, but the viewer does not have sufficient download speed the bitrate will not change.

HOW TO DELIVER MULTI-BITRATES: LIVE

For live streaming, there are two approaches for delivering multi-bitrates: sending multiple bitrates via the encoder or using cloud encoding to create additional bitrates from a higher resolution feed.

The first method is supported by a variety of encoders, although the methodology of achieving this differs from encoder to encoder. Flash Media Live Encoder and TriCaster, for example, need a “%i” amended to the end of the Stream Key. This will tell the host that the stream might be sending multiple streams/bitrates. Other encoders, like Wirecast, mandate that each bitrate be defined with its own unique number. This is done by simply adding a number to the end of the Stream Key, ex 1, 2, etc.

The downside to the method of sending multiple bitrates from the encoder is that it requires additional CPU and, more importantly, additional bandwidth from the upload speed as well. Additional bandwidth is needed because the broadcaster is effectively sending multiple streams. Under normal conditions, you want to have an upload speed that is double your proposed video bitrate quality. So a 2Mb per second feed would require a 4Mb per second upload speed. However, if you are doing multi-bitrates, the demand on your upload speed increases. For example, doing a 2Mb per second feed and a 1Mb per second feed would require an upload speed of 6Mb per second, as you are adding together all the bitrates you are streaming at (double the total of 3Mb per second that you are sending.)

The second method is from doing cloud based transcoding. This involves sending a single high resolution, high quality bitrate feed that is then downscaled to offer viewing options for viewers with lower internet speeds.
KEYFRAMES AND ADAPTIVE STREAMING

When compressing video content, a keyframe (also called an intra-frame or i-frame) is known as the full frame of the video image. Frames that follow, called either predictive frames (p-frames) or bi-directional predicted frames (b-frames, not available in baseline streaming), only contain information that has changed between the keyframes. The keyframes are found at regular intervals during the video content, which are set inside the encoder. As a result, more keyframes generally means more bandwidth is required or that the resulting video file will be larger.

When dealing with content that can be adaptive, it is recommended to use a keyframe interval (sometimes called keyframe frequency on encoders like Adobe Flash Media Live Encoder) that has a short duration. The reason for this is that the change of a bitrate can only occur during a keyframe. So for example, if the system detects that they should be on a lower or higher bitrate and the keyframe interval was set to 10 seconds the viewer might have to wait a full 10 seconds before the change is actually made. The keyframes across the various streams also have to be at a predictable interval. So auto keyframe options should not be used over a predictable, per second interval.

Ustream recommends 2 seconds for the keyframe interval and the keyframes have to be present in the same location or the switch will not function.
CLOUD ENCODING: USTREAM APPROACH

Found as a component of Ustream’s Broadcast Settings, Multi-Quality Streaming allows for the creation of multiple bitrates from a single stream through cloud transcoding. This process is dependent on the resolution of the source stream. For a lower bitrate and resolution option to be created, the source given has to be at a greater resolution. These encodes will, by default, be in sync with the keyframe interval of the source. If the keyframe interval is not frequent enough, for example: keyframe intervals considerably longer than 2 seconds, then extra keyframes will be inserted into the encoded versions. Lower bitrates are created in the formats of 240p, 360p and 480p.

THE EXACT SPECS:

<table>
<thead>
<tr>
<th>RESOLUTION</th>
<th>VIDEO BITRATE</th>
<th>AUDIO BITRATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>240P</td>
<td>426 x 240</td>
<td>500 kbps</td>
</tr>
<tr>
<td>360P</td>
<td>640 x 360</td>
<td>750 kbps</td>
</tr>
<tr>
<td>480P</td>
<td>854 x 480</td>
<td>1000 kbps</td>
</tr>
</tbody>
</table>

Note: 360p and 480p options are available on Enterprise and Corporate plans only. The 240p option can be found on all Pro broadcasting plans.

Sending multi-bitrates from the encoder is supported on desktops using the Flash player. This process is not supported by mobile players using HTML5 players. However, the HTML5 players do support cloud transcoding.

RETAINING VIEWERSHIP AND SERVING HIGH QUALITY CONTENT

This paper has laid out why it’s important to avoid buffering, as viewers are more likely to abandon content based on its frequency. That said, broadcasters have a critical desire and need to deliver high quality video content. Adaptive bitrates offer an opportunity to do both, allowing broadcasters to serve high resolution content to those who can view it while not sacrificing an audience that lacks the bandwidth to watch it.


Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Apple, iPhone, iPad, iPod touch, and iOS are registered trademarks or trademarks of Apple Inc., in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.