

eBook

# Adaptive Streaming: Improve Retention for Live Content



# Using adaptive streaming to improve retention for live content

---

## **Introduction**

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. An end product that can reach people with HD (High Definition) content on robust setups, while still being able to deliver content without the threat of near constant buffering to those on slower setups.

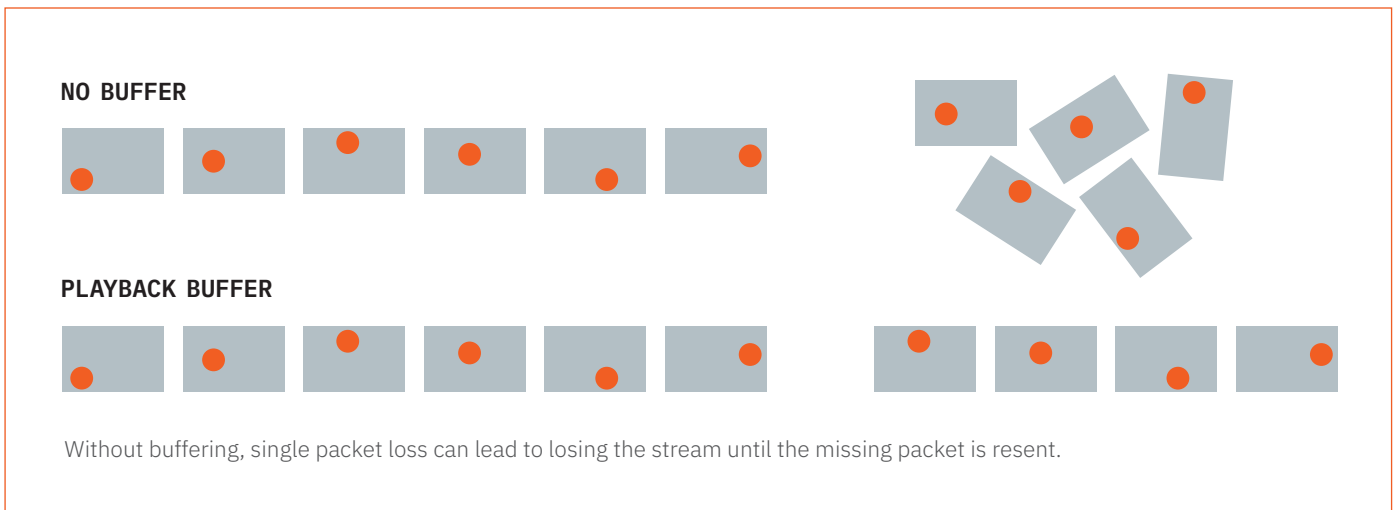
## Understanding Playback Buffering

Video streaming is achieved by sending over packets or video chunks of data. Without any added measures, if a packet or video chunk is lost, the video player application would stop receiving data until the lost packet was received. This is an issue as later packets will continue to be delivered. For example, if packet 50 was lost, the video would stop until it's received. This would occur even as later packets, like 51-75, arrive. Once 50 is finally received, packets 50-75 would be delivered all at once, causing a disruption in the video until this event takes place.

To prevent 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 generally wait until more data is received to continue playback. Generally the player should keep the video stopped for longer than is necessary. This is done so the player can fill up the buffer before playback resumes. Without this process, the player would likely buffer again quickly once a missing packet or video chunk was received.

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



## Business Impact of Buffering

For almost all use-cases, avoiding buffering should be the goal. The reason for this is that the viewers tolerance for buffering is already very small and continues to decrease. A [2016 IneoQuest study](#) says over half of all Americans experience “Buffer Rage.” The study sites that 34% of respondents under the age of 35 suffer from Buffer Rage more frequently than road rage. Additionally, it was found that 58% experience their highest level of stress when buffering occurs while they’re using their mobile device<sup>1</sup>.

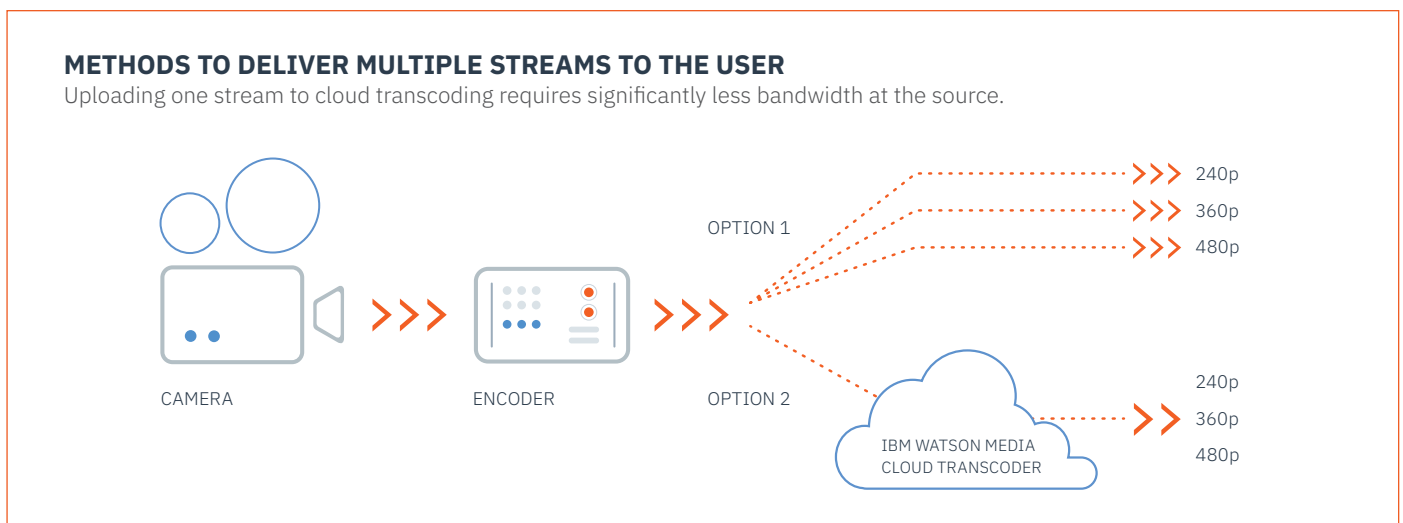
The level of impact also depends on the type of content being streamed. For example, a [study by Conviva](#) found that the average viewing time for sports content was 40 minutes in HD, but dropped to an anemic 1 minute if the viewer encountered buffering<sup>2</sup>.

## 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, usually 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. IBM Watson Media 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.

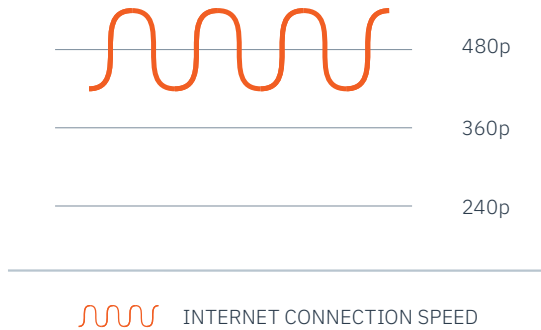


**OPTION 1:** Generate multiple quality streams from the source & upload them to the cloud.

**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 many setups. The system can effectively adjust the bitrate to match the optimal result to the download connection attempting to view the content. Usually 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 easily rationalized.

### PLAYBACK WINDOW SIZE

A system can also monitor playback size in order to determine what bitrate and resolution combination 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.

IBM Watson Media utilizes a combination of the viewer's connection and the playback window size in determining what bitrate to serve.

## Adaptive Streaming: Changing Bitrates Mid-Stream

Setups make 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 settings. For example, if the viewer's network connection strengthens or the playback window size is changed.

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.

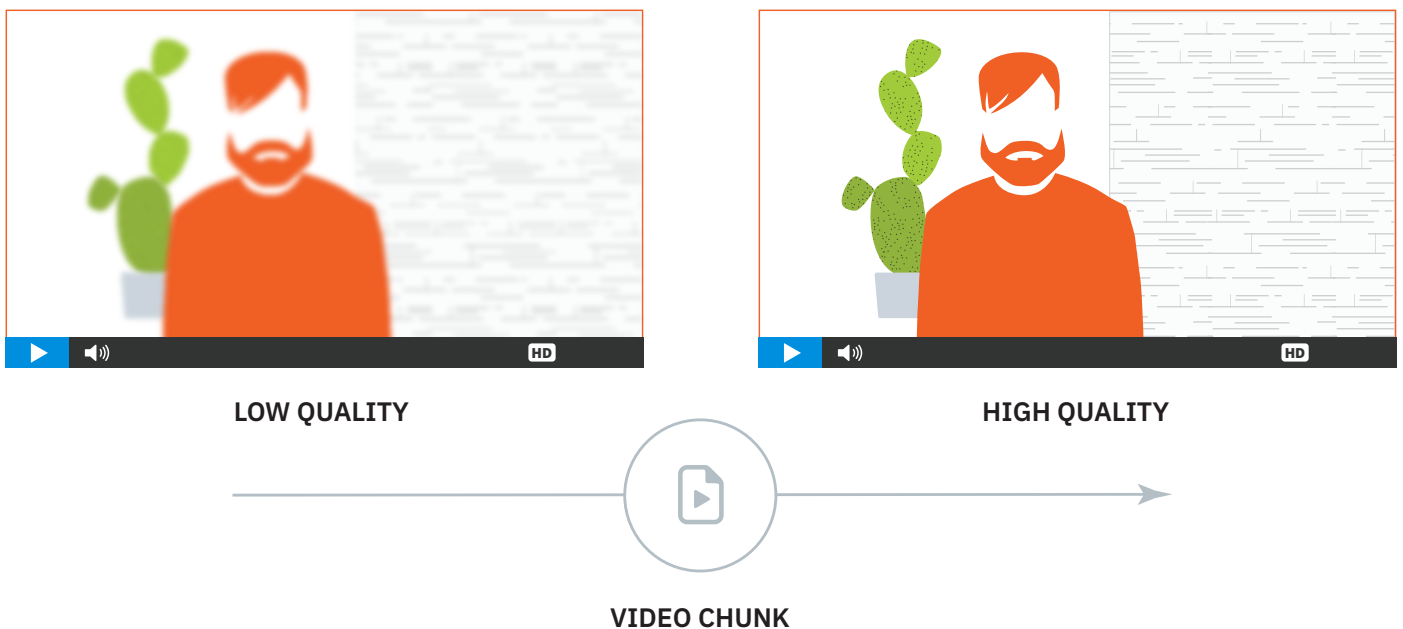
IBM Watson Media offers both the ability to change bitrates mid-stream and also checks the buffer to make this determination. While the first video chunk is loaded, the reporting begins in the background with each downloaded chunk. The system projects available bandwidth from the viewer while they watch 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. This said, if the viewer manually changes the bitrate the player will not change their selection. On the note of manual bitrate selection, these options are generally presented to the viewer in terms of available resolutions.

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.

### RAPID FEEDBACK

allows for larger chunk download resulting in a higher quality stream



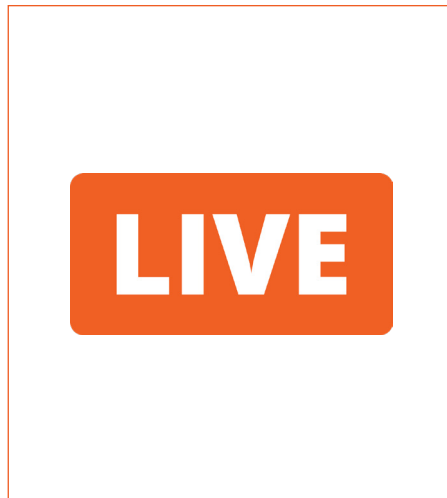
## Conditions that Trigger a Bitrate Change

Incidents that can trigger the change in bitrate from an IBM Watson Media broadcast include available bandwidth (download speed), playback window size and the performance of the buffer.

In terms of available bandwidth, as previously mentioned this process 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 the speed to be tested at a rate that is a certain percentage higher than the bitrate of the video content.

For the playback window size, if this 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 might be a 720p version if available. This detection is precise, meaning if the playback window size was exactly 20%, like in the example above if it was 576 pixels instead of 577 pixels, it will not make the change. The change in window size can take a few seconds to register, and will change at the next keyframe.

Naturally, both of these conditions have to be met to trigger a change. For example, if the playback window size is 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.)

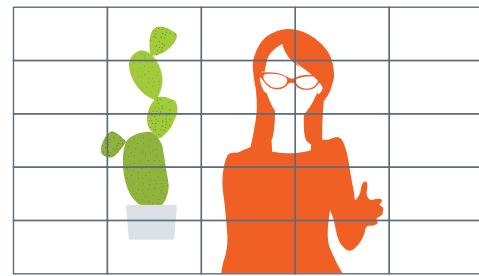
The second method is from doing cloud based transcoding. This involves sending a single high resolution, high quality bitrate feed that is then downscaled to offer viewing options for viewers with lower internet speeds.

## Keyframes and Adaptive Streaming

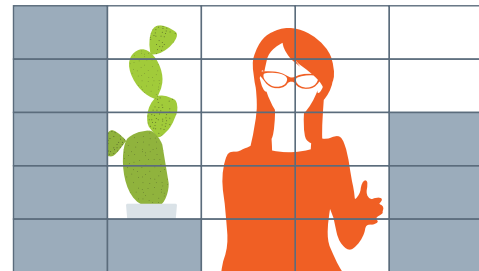
When compressing video content, a keyframe (also called an intra-frame or i-frame) is known as the full frame of the video image. Frames that follow, called either predictive frames (p-frames) or bi-directional predicted frames (b-frames, not available in baseline streaming), only contain information 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.

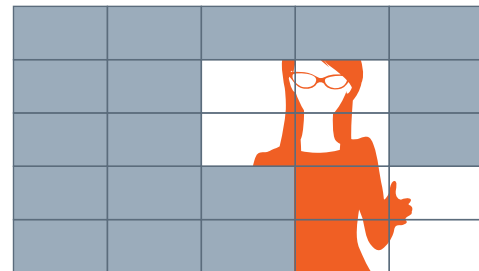
IBM Watson Media recommends 2 seconds for the keyframe interval and the keyframes have to be present in the same location or the switch will not function.



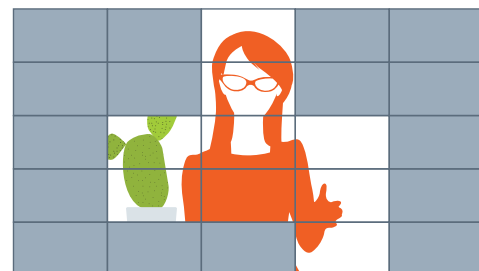
**KEY FRAME**



**DELTA FRAME**



**DELTA FRAME**



**DELTA FRAME**



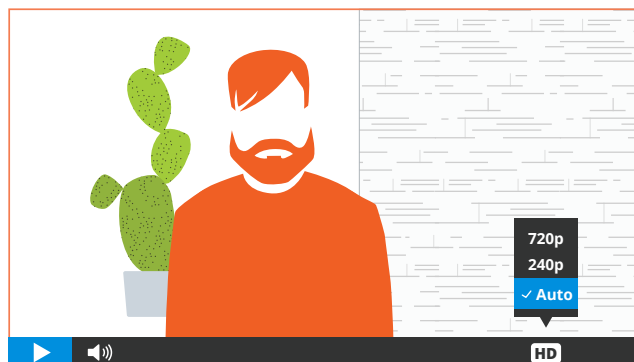
## Cloud Encoding: IBM Watson Media Approach

Found as a component of IBM Watson Media'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 and 720p.

### THE EXACT SPECS:

	RESOLUTION	VIDEO BITRATE	AUDIO BITRATE
240P	426 x 240	500 kpbs	56 kbps
360P	640 x 360	750 kpbs	64 kbps
480P	854 x 480	1000 kpbs	64 kbps
720P	1280 x 720	2250 kpbs	64 kbps

For sending multi-bitrates from the encoder, this is supported on desktops using the Flash player. This process is supported by HTML5 players as well, but only if the incoming MBR (Multiple Bit Rate) stream uses H.264 as the video codec and AAC as the audio codec. Additionally the HTML5 players do support the cloud transcoding. While this article mostly focuses on live content, in relation to adaptive delivery, this process is also available on VOD (Video On-Demand) content as well.



## RETAINING VIEWERSHIP AND SERVING HIGH QUALITY CONTENT

We've laid out why it's important to avoid buffering, as viewers are more likely to abandon content based on its frequency. That said, there is an understood desire 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.

## About IBM Watson Media

Created in January 2016, IBM Watson Media brings together innovations from IBM's R&D labs with the cloud video platform capabilities of Clearleap and Ustream. Through the unit, IBM delivers a powerful portfolio of video services that spans open API development, digital and visual analytics, simplified management and consistent delivery across global industries. IBM Watson Media supports top media and enterprise companies with reliable video on-demand and streaming services.

For more information on IBM Watson Media, please visit [www.ibm.com/watson/media](http://www.ibm.com/watson/media).



---

© Copyright IBM Corporation 2019

IBM Cloud Video  
San Francisco, CA 94105

Produced in the United States of America  
June 2019

IBM, the IBM logo, IBM Watson Media logo, [ibm.com](http://ibm.com), and Ustream are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.

A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [ibm.com/legal/copytrade.shtml](http://ibm.com/legal/copytrade.shtml)

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

This document is current as of the initial date of publication and may be changed by IBM at any time.. Not all offerings are available in every country in which IBM operates.

The information in this document is provided "as is" without any warranty, express or implied, including without any warranties of merchantability, fitness for a particular purpose and any warranty or condition of non-infringement. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

Statement of Good Security Practices: IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed or misappropriated or can result in damage to or misuse of your systems, including to attack others. No IT system or product should be considered completely secure and no single product or security measure can be completely effective in preventing improper access. IBM systems and products are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM does not warrant that systems and products are immune from the malicious or illegal conduct of any party.

---

## Footnotes

1. Stop The Madness! Buffering Is Driving Us Crazy, P.J. Bednarski, MediaPost, March 17, 2016, <https://www.mediapost.com/publications/article/271503/stopthe-madness-buffering-is-driving-us-crazy.html>
2. Bookman, S., Buffering time decreases 12.4 percent, but online video viewers losing patience, FierceOnlineVideo, FierceMarkets, April 3, 2014, <http://www.fierceonlinevideo.com/story/buffering-time-decreases-124-percent-online-video-viewers-losing-patience/2014-04-03>