Best Practices for Multi-Device Transcoding (2018 Edition)

Last Modified on 10/08/2019 7:49 pm IDT

by David P. Kronmiller

Introduction

Streaming video has become the way of everyday life. People now have grown accustomed to being able to consume video regardless of where they are, who they are or how busy they might be. It is an on-demand world and content reigns supreme.

In 2011 when this article was first written, video was primarily consumed by desktop users and early smart phone users at lower quality. We now, in 2018, can record 4K Video at 50 Mbps on our phones and stream not only our favorite shows on our devices but we can immerse ourselves in virtual reality and 360 videos.

This article outlines the Best Practices for Multi-Device Transcoding covering both basic Transcoding concepts but also the general specifications for different Device Types, Viewing Conditions, Delivery Methods and Container Formats.

STREAMING CONTENT

Content flows to an end user's system in a stream of data much like a river might flow from a mountain reservoir to a small town. All content that is viewed on the internet inside an application is considered streaming content.

IT'S STILL ABOUT BALANCE

The challenge of streaming video is still to find the right balance between bitrate and resolution as it relates to an end user's connection speed and system capability. Though connectivity is improving globally, there are still a variety of factors that can affect a video's ability to playback smoothly including: the specific network they are on, the device they use and how they are using their device.

BANDWIDTH

Though bandwidth is faster than ever, it is not an even landscape worldwide. According to Akamai's recent 2017 report on connection speeds currently South Korea has the fastest on average connection speed of 28.6 Mbps - the United States by comparison has just made it's way into the top ten fastest nations with 18.7 Mbps. That is a 10 Mbps spread!

The global average per Akamai's report is 7.2 Mbps.

This means that the transcoding solutions for South Korea would be different than those for say Brazil which has an average of 6.8 Mbps. This impacts not only bitrates but also the resolutions that are possible to stream in those regions. Higher resolutions demand higher bitrates to maintain good image quality - so though South Korea might be able to stream 4K, Brazil might only be able to handle 720.

Adding challenge to this balance is that some ISP's and even some countries can cap or limit bandwidth on top of the overall limit of the region. Meaning someone with a good 20 Mbps connection plan might, at certain points in the billing cycle, get capped to something much lower, like say 10 Mbps or even as low as 1 Mbps. This is especially true of mobile consumers on smart phones.

And connection speed is just one variable that might impact a users' streaming experience.

NETWORK

The average user will connect to the internet over their home WiFi or office network. This home network is shared by everyone in the household and the number of devices attempting to connect to this home network can affect the overall bandwidth.

For example, in this white paper writer's home WiFi network currently, I get a solid 50 Mbps download speed with only the main living room HD television streaming Netflix and one computer browsing the internet. However, if a second TV starts streaming, say on my home office ROKU where I frequently stream Food Network, my download speed plummets to 36.7 Mbps. If my wife then starts streaming on her iPhone 7 our download speed bottoms out to 13.6 Mbps. All good connection speeds but it illustrates how a couple of devices can change the overall network conditions on a home WiFi network - even a fast network like mine. Interesting to note - if that computer that was already connected to the network started also playing HD content the network plummets to 9.57 Mbps.

Now imagine that same scenario in a household that only has an overall bandwidth of 7.2 Mbps.

This problem of network sharing is even more challenging for corporations and universities where a larger number of people are trying to access the network, not just the wife and kids. This means that the solution for a premium media company will be vastly different than the solution for a University streaming class lectures or a corporation who might stream internal training videos over their building's broadband. In a shared setting like an office or college campus, when WiFi is used, not only might there be a large number of actual users accessing it at the same time, but the building's physical layout could mean one person might be closer to the WiFi and another might be in the corner of the office that barely gets a good signal yet both must stream the same video.

DEVICE

A viewer may be accessing the content on a variety of devices - each device has it's own advantages and disadvantages and potential ways by which the device may be limited. We will get into specific device settings later in this document.

AGE

The device's age can play greatly into it's ability to handle streaming content. An older device may not have the hardware, memory or OS to handle HD content - a user with an older device might rely on lower bitrates and resolutions being available.

BROWSER

The browser they are using may not be compatible with the streaming options available for a given video, though this is becoming less of a problem with the adoption of HTML5 and MPEG DASH delivery. The Browser might also just need an update, or the update may have limited certain playback scenarios- for example, many browsers have now limited and turned off Adobe Flash delivery as it's being phased out and has security risks.

USAGE

Another factor that might limit a device's ability to stream video is how many other programs or Apps are open on the device at the same time. Even having multiple tabs open in a browser can greatly limit a systems CPU. A FB page or

YouTube page in an unopened tab is still draining resources – the more images and video on a page, the heavier the overall drain.This problem is easy to solve as the user just needs to close the applications or browser windows they are not using.

A BRIEF HISTORY OF ENCODING

Current encoding methodologies got their start 30 years ago at the advent of digital video in television production and was at first limited by traditional broadcast standards for televisions receiving their signals over an antenna and being displayed on a 4x3 interlaced CRT (aka a square television) and the limited data rate of a Compact Disc (CD) of 1.5 Mbps.

It all began with the H261 Codec put into use in 1988 and created by the Video Coding Experts Group. H261 would go on to become the first standard for video conferencing but resolutions and bitrates were limited (resolutions 288 and 144 with bitrates only up to 2 Mbps possible.)

Also, in 1988, the Motion Picture Experts Group, or MPEG, formed to tackle digital video standards as a consortium of experts came together to set the stage for streaming video's explosion 20 years later. The MPEG standard would adopt H261 as its first codec and become the basis for the standard that is still used today.

Each standard MPEG would release improved on the previous one in significant ways, adding profiles and levels that could be adjusted, de-noising filters, and ultimately introducing B-Frames and P-Frames, allowing further compression.

GENERAL SETTINGS and CONCEPTS

Before we can dive into the specific device groups and their intended settings it is important to understand the essentials of transcoding. This section will detail the most commonly used settings and terminology for transcoding.

ENCODING

First let's start with the basic idea of video encoding. Video encoding is taking a source moving image sequence and compressing it into a format that is then readable, or decodable, by an end player or set of players. The reason I use the heavy phrase "moving image sequence" is that a video or film really is simply a series of still images playing back at a certain speed (frame rate). In order to take, for example, an old fashioned 35 MM film, and put it into a computer for editing, the film has to be scanned one frame at a time and the information in each 35MM frame translated into 1's and 0's – that translation is encoding.

TRANSCODING

Now "transcoding" is taking a previously encoded piece of video and translating it further into another format or process.Transcoding will convert the source file into one or more newly and more compressed streams that can then be played in a player on a computer or mobile device depending on the settings and methodologies used.

USE CASE: A video editor has 4K Source footage @100 Mbps from the camera used to capture the content but the 4K video is too large in resolution and bitrate for the video editor's editing system to handle and the video editor would prefer to be able to edit in 1080 instead. The video editor could convert that 4K source footage by transcoding it into a new proxy file that is 1080p at a much more compressed 7000 Kbps with a 1 second GOP or Key Frame Interval. The video editor can then, once done editing, relink the original 4K files and output a 4K source.

BIT RATE

A Bitrate is a measurement of data speed across a network, often in Kilobits per second or kbps (1000 bits per second). This number correlates with potential bandwidth levels that a user may experience and should be in balance to the resolution of the stream.A household who has a data plan limited to 10 Mbps cannot handle a bit rate over 6500 kbps. You may wonder why if that household can handle up to 10 Mbps they would only be able to handle 6500 kbps, why not the full 10 Mbps?

The answer is because though the average data rate may be 6500 kbps for that video stream, it will spike at least 30% up to 50% of the average at various points in the video stream if the content creator has transcoded using variable bit rate, which is a common method. Additionally, that home bandwidth is limited by other devices using it and the player or device displaying the video might have additional plugins/widgets etc. such as analytics tools & DRM (Digital Rights Management) that might add additional overhead. Therefore, a buffer must be considered.If you are using Akamai HD HTTP Streaming bit rate spikes are less of an issue as it uses client-side caching allowing for a higher threshold – however if network conditions worsen those spikes may still present a problem. Again, it is always about a balance between performance and visual quality.A video encoder can choose between a few different methodologies for managing bitrate. These are:

- **Constant Bitrate:** the video bitrate does not change regardless of the image complexity.

- **Variable Bitrate:** the video bitrate fluctuates depending on the specific complexity of changes from one frame to another. If there are few changes from frame to frame, those frames can be predicted and compressed easier, allowing a lower bitrate for those sections. If there are big changes from frame to frame, say for a movie trailer, a higher bitrate or level of complexity may be required to avoid visible quality artifacts.

AVERAGE vs MAX

The Video Bit rate has two main components: The Average and the Max

- **AVERAGE:** The average bit rate for video should coincide with the target bandwidth of the end user and should be in balance with the resolution.
  Just changing a bit rate alone is not sufficient for dealing with bandwidth limitations and is not recommended. It is advised that low bitrates also scale down in resolution so that the end user has a good balance between image and playback quality.

- **MAX BIT RATE:** A Max Bit Rate governs the ceiling that a variable bit rate may reach and should be within balance of the average and the target connection/device. Max Bit rates do not effect Progressive Download and some services like Akamai have some built in functionality that limit the impact of bit rate spikes. However, not everyone uses Akamai, and Progressive download is not often ideal for streaming.

Potential performance issues due to High Max Bit Rate include: Unwanted Bit Rate Switching, Stutter, Player Crash and Buffering.

Industry standards say you should calculate the Max bit rate by taking the Average and adding 50%. It is recommended to reduce this even to 30% to create a truly consistent stream but still taking advantage of a variable bit rate.

USE CASE: If a transcoded stream has an Average of say 1400 kbps but spikes at 2600 kbps that user may experience one of the above performance issues if their bandwidth can only support between 1000 kbps and 2000 kbps, which is highly likely for many users who's ISP's cap their data plans. This would mean that when the stream spiked to 2600 it is outside the threshold that the user can handle.

BITS PER PIXEL

Bits per pixel is a measurement of how many bits are assigned to each pixel in the encoded stream. The higher the number the better the image quality and accuracy as compared to the source file color and sharpness. But remember visual quality is also qualitative and subjective. Our eyes are only capable of seeing so much detail, which is why compression works so great. Our eyes fill in the ridges and smooth over any encoding visual artifacts such as blocking. Also other features used to encode the video may mask any quality issues –such as when noise reduction is utilized.

BUFFERING

A big attribute of streaming video is the Buffer. The Buffer is where data is held until it is needed. The way streaming video works over an HTTP type connection is the video and audio are coming into the player inside packets of data. These packets stream into the buffer and then as a player needs them they are pulled from the buffer and displayed. The rate at which this happens is determined by the encoded streams buffer settings. It is recommended that the buffer be set to 150% of the bitrate. So, if the average video bitrate is 4000 Kbps, the buffer should be at least 6000 Kbps, or 1.5 seconds, though you can go higher with newer devices, especially smart TV's or connected devices which may support up to 2 seconds or higher.

Think of the buffer as a cup and the player as a very thirsty runner. If the thirsty runner, or player, goes to drink from the cup and there is no data in it, then the thirsty runner might trip and fall down from lack of good hydration aka a player crash or simply stumble, or buffer, a little until more water can be found in the cup. When the cup is empty, but the player is thirsty this is called a Buffer Underrun.Now you might fill that cup too quickly, and it might start to spill over, losing valuable data, causing buffering or stuttering issues as well – this is called Buffer OverRun. So, you want that flow into the cup to be in balance with the needs of the playback and always be at a steady pace.

CODECS

Codec is short for Code/Decode and refers to the methodology and encoding library used to transcode the video or audio. For a stream of video or audio to be played back the receiving player must be able to decode the video and present it properly.

H264

For streaming video, the primary codec currently in use is called H264 (aka Advanced Video Codec or AVC). H264 is the descendant of the MPEG codecs used for Video Recording, DVD and Broadcast. MPEG stands for Moving Picture Experts Group and is the consortium of industry professionals who have created the MPEG standard and methodologies for distributing digital video. H264 is more specifically referred to as MPEG-4 part 10 when discussing its development. A license free version of the H264 standard is X264 - an open source version that meets the MPEG standard.

H264 is broken down into 3 different profiles of complexity: Baseline, Main and High. Each profile is further divided into Levels that govern the intensity of certain functionalities and features and allow larger bitrates and resolutions as the level increases. These settings are crucial for good device playback as not all devices support all the various levels and combinations. The differences in the profiles comes down to the maximum resolution and bitrate it supports and certain predictive features such as allowing B-Frames.

- **BASELINE:** Baseline is the original profile level back when streaming video was first being used primarily for video conferencing. It does not support VBR and B-Frames and therefore will have less visual detail than Main or High profiles and will contain an increase in encoding artifacts.
- **MAIN:** Main is the next profile level utilized. It includes better motion prediction. B Frames and Automatic Scene

Detection.

- **HIGH:** Even better color accuracy, motion prediction and detail. Harder to decode.

LEVELS

The level utilized determines how high of a resolution and bitrate is possible as well as other features that get turned on as the level increases. They range from Level 1 through Level 6.2. Here are some nominal settings for Profiles and Levels as they relate to resolution:

- 360p Baseline L3
- 720p Main L3.2
- 1080pHigh L4.0
- 2160p High L5.0

Not all devices handle higher levels and profiles, therefore it is important to check device specifications for proper balance and limits of these settings. Some of this are discussed in the Devices section in this article,

H265

The newest codec being utilized for streaming is H265/HEVC, the next generation of streaming codecs. HEVC stands for High Efficiency Video Coding and like the name suggests is all about improving compression compared to H264. It is not widely used yet for streaming, though is thought to be used more starting this calendar year of 2018. It specifically handles the larger formats 4k and 8k much better.The qualitative difference however is noticeable. Here are two images, both from 400 Kbps Variants at 360 Resolution:

H264

H265

Notice that despite the low resolution, the H265 maintains color and contrast and notice how the actor's head and hair is visible in the H265 but blends into the rocks in the H264.

H264

H265

H265 also has the added benefit of smaller file sizes and lower bitrates for same quality. This can mean storage savings and much more flexible bandwidth.

STREAMING METHODS

There are two primary streaming methodologies Progressive Download and Adaptive Streaming.

PROGRESSIVE DOWNLOAD

As the name might suggest progressive download is when the video being viewed has to be downloaded to the user's

computer prior to playback. Though not as used today as it once was at the beginning of streaming video say 10 years ago, it is still useful for when a good internet connection is not available. In the original days of streaming when a user played a video on a website inside a player, the player would first have to download a certain percentage of the file first before playback began. If the file was large or of a higher resolution the player might buffer for a little bit waiting for another portion of the video to download. Typically, a user might select a video, pause it and walk away to wait for the whole video to download first so that playback would be smooth. The limitation of this method is of course flexibility and consistent playback and in ability to quickly seek/scrub.

ADAPTIVE STREAMING

Both the antiquated Flash Player and the now widely adopted HTML5 Video/MPEG DASH standards utilize Adaptive Streaming, whereby a player or application will monitor the network traffic and bandwidth of the user, determine also what device the user is using and serve up an appropriate video and audio file that matches best to those changing conditions. As conditions fluctuate (say on a mobile device over WiFi) the player/application requests a matching file, or VARIANT. This allows a user to stream video under a variety of network limitations without creating a huge load on the end user's device.

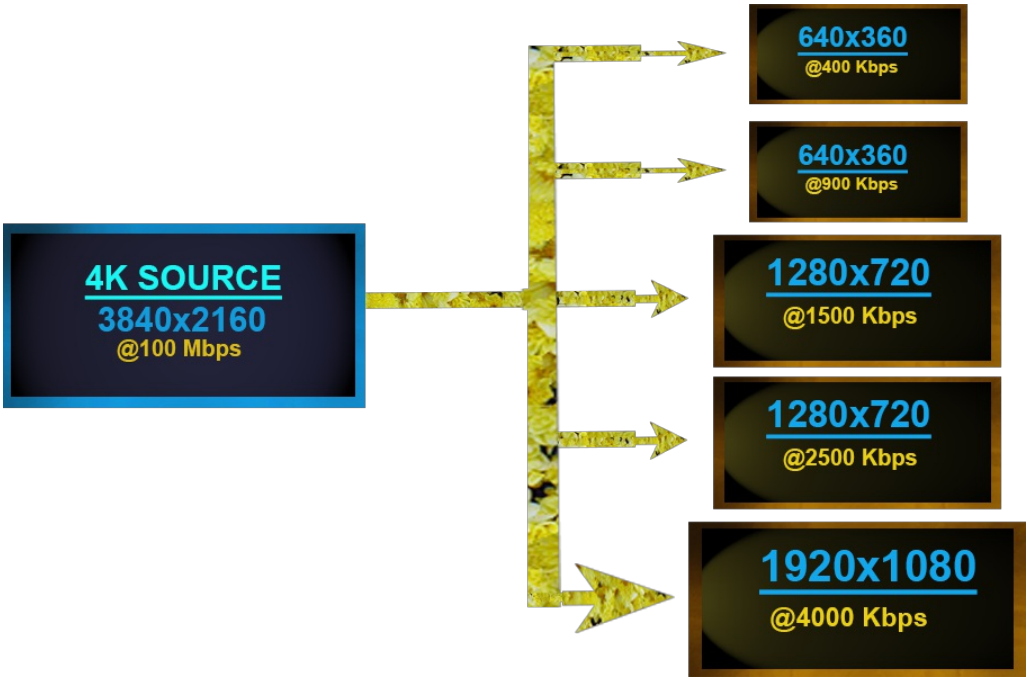
ADAPTIVE SET

An Adaptive Set is a package of transcodes for the same video that span multiple bit rates and are meant to find a balance between connection speed and resolution. Every video you watch on the internet most likely was transcoded into 6-8 different MP4 files of various resolutions and bitrates.

For Adaptive to work, all the streams in an Adaptive Set must be in alignment. Each variant should have the same GOP/Key Frame Interval, audio sample rate and ideally video frame rate. Only resolutions, bitrates and profile levels should change.

Each video is transcoded into a variety of variants (for Kaltura we call them Flavors). Each variant represents a bandwidth level and resolution that incrementally scales down from a 1080 or 4K top level down to a low resolution 240p at 100 Kbps level or even lower. Apple iOS spec, for example, recommends including an audio only stream for the lowest level in order to preserve the stream and keep the player from crashing, requiring a player restart by the user.

What adaptive streaming does is switch which variant you might be seeing at any given time to one that fits your bandwidth and devices capabilities. So if you're on a mobile device, you might see your video go from high quality down to a lower quality as your mobile or WiFi signal fluctuates.



CONTAINER FORMATS

When most refer to a video or audio file what they are really referencing is what is called a container format. A container holds the metadata, video and audio streams along with subtitles or timecode when needed. It can often contain a variety of video and audio formats - for example a QuickTime Container format .MOV could contain 1 video track using H264 and 1 audio track using AAC or it could contain 1 Video Track using ProRes (Apple Codec) and 1 Audio Track of Apple Audio (ALAC Codec).

It is important to differentiate between the .MP4 container and an MPEG-4 Codec. Many confuse the two terms and think that .MP4 is a codec - it is not. The actual MPEG-4 Codec (as discussed in the Codec Section) is also called H264 or AVC (Advanced Video Codec).

Streaming video today deals with two primary container formats, MP4 and WebM.

MP4 Container is for H264/AVC encoded content often with AAC audio. WebM is a Google format that uses VP9 video codec and Vorbis/Opus Audio codecs. It is the primary codec for YouTube given Google/Alphabet is YouTube's parent company. Both formats result in good compression, adaptive switching and solid visual quality.

Now Source files could be formatted in a wide variety of containing formats. We will list some of the main ones used below. Containers can have multiple video and audio streams and it's important to understand your source format/layout prior to creating your transcoding work flow. Often studio level source files use .mxf or .mov containers with ProRes or DNxHD video codec and multiple audio streams.

A typical studio level source file formatting might go as follows:

…among others.

STREAMING CONTAINERS:

SOURCE CONTAINERS:

…or it could be far more complex:

RESOLUTIONS

The resolution of a video is measured in the height in pixels. A content creator could use any resolution, however there are industry standards to consider.

Resolutions are affected by two different display methodologies – Progressive Scan vs Interlaced. Interlaced means that each frame of video is split into alternating rows of pixels, this was to allow a video image to fit inside the frequencies used for broadcast over the airwaves via antennae. So one image would be split into Row A and Row B – all of the Row A's combine to form Field 1 and all the Row B's become Field 2. These two fields are scanned into the receiving television in an interlaced fashion, A then B, then A then B until all 525 lines of horizontal standard definition resolution are filled for a single frame. It essentially compresses the data so that the entire image isn't presented in one packet but rather 2 smaller ones being fed to the television alternatingly. HD television broadcast still utilizes this methodology by using 1080i where "i" stands for interlaced.

Progressive Scan does not split a single image into 2 fields that are combined, rather it presents each line of pixels in

sequence starting at the top of the frame and proceeding down to the last row. This results in a sharper image and better overall quality.

All major resolutions are based on incrementally doubling each from the last. For example, HD must be 2x the resolution of SD, so therefore 525x2=1080 – of course there are resolutions in between these bench marks as well.

- SD : 525 (and lower)
- hd : 720p
- HD : 1080p HDTV : 1080i 4k : 2160p
- 8k : 4320p

MULTIPLE OF 16

Encoding is done by breaking an image down into 4x4 or 16x16 blocks of pixels. So every resolution, in order to be most efficient, should be a multiple of 16. This is often shortened to the more hip sounding, Mod-16.

UNIQUE FRAME SIZES

Certain web applications of video might call for unique resolutions and frame sizes. Facebook video for instance uses a square vide frame of 1080x1080 or an aspect ratio of 1:1. It is not recommended to output these unique web-based frame sizes in an transcoding engine but rather from a non-linear editing system where proper matting and cropping can be done.

RESOLUTION vs DIMENSION

It is easy to get resolution and dimension confused – primarily because they often refer to the same thing. However, Resolution refers to the amount of detail the image has and Dimension only refers to the Aspect Ratio of that image. Both are measured in Width and Height.

Resolutions should scale according to the desired device and bitrate. It is advisable to not try to scale video up from the source dimensions but rather present it in the original resolution of the source master. Scaling up will result in noticeable degradation in picture quality.

Here are some common Theatrical and Broadcast Aspect Ratios:

- 133/1 – 4x3 Television
- 178/1 – 16x9 Television and the most common Film Aspect Ratio
- 185/1 – A Common Film Aspect Ratio
- 166/1 – An older Aspect Ratio that was often used in Animation 235or240/1 – Next to 178, the second most common Film Aspect Ratio

NOTE: NOTE: An Aspect Ratio is calculated by dividing the width by the height. 178 is actually 1.78 to 1 as a ratio of that width to height. A Resolution of 848x480 using this formula, for example, comes to 1.766 or rounded up, 1.78 – tradition drops the decimal and so this is simply called 178. As you calculated your resolutions keep this formula and the need to have the resolutions be divisible by 16 in mind.

GROUP OF PICTURES and I-FRAME INTERVALS

The key to video encoding is the use of compression – taking a large resolution image as it might appear on 35mm film or HD or 4K video and making it viewable by a home user. Though bit rates and resolutions do much of the work to create a balanced viewing experience there are other factors that also contribute to this compression.

The most vital of these is the I-Frame Interval, sometimes referred to as the Key Frame Interval or the Instantaneous Decoder Refresh (IDR) or the GOP (Group of Pictures). All refer to essentially the same thing – the need for a reference frame from which guesses can be made about the subsequent frames. Every frame of video of a master once transcoded is not fully represented.As described previously, this all about “moving image sequences”.

Video is broken down into small chunks of images known as a Group Of Pictures aka GOP. Each Group of Pictures starts with an I--Frame/Key Frame.Every encoder can call out how far apart key frames are created – this is referred to as the Key Frame Interval. The frames in between Key frames are called P- Frames (predictive coded picture) and B-Frames (bi-directionally predictive).

P frames make a guess as to what the picture looks like based on those Key Frames. B frames are additional reference frames that help improve the quality of P-Frames and the overall look of the stream.Ideal Key Frame intervals should be between 2 and 10 seconds. Meaning, for a 29.97 fps piece of video, the key frames should be between 60 frames and 300 frames. 2--3 seconds is standard for iOS and Mobile applications and is widely used for Desktop and Connected Devices as well.B--Frame usage should be limited to 1 to 2 reference frames – going over 3 reference frames may cause poor playback on some players (QuickTime for example).

However, if you are confident that your users will be using players that support B--Frame decoding (Flash, HTML5, HTTP Live Streaming) you can increase these B--Frames to increase picture quality. Understand though that you will be increasing file size which may slow down load times and cause some buffering.IMPORTANT CONSIDERATION: Key Frames are also Adaptive Switch Points – an adaptive player will switch bitrates when needed at Key Frames.

If the GOP is too long a user’s player may crash if a switch is required because the user’s bandwidth has dropped prior to reaching the next IDR frame which acts as a switch point. Also if a Key Frame Interval is too close together file sizes increase and the wrk needed to decode the picture increases causing a poor playback experience, especially for users with mid to low range bandwidths.

The only time you should use a key frame interval of 1 frame, where every frame is a key frame, is for mastering or if the resulting file is meant to be used in a non--linear editing program.

FRAME RATE

A variety of frame rates can be used for streaming video. Frame Rate refers to the frequency that a new frame of video is presented each second during playback.

The frame rate can go as low as 12 fps (Frames Per Second) or up to 60 fps. Normally you want all your variants to have the same frame rate so that when the player dynamically adapts to the network conditions and switches to a different variant/flavor asset there is no noticeable skip. Also transcoding engines do not do frame rate conversions very well, meaning taking a source file that is at 30 fps and dropping frames to create a streaming variant at 15 fps might result in noticeable frame skips during playback.Some choose to reduce frame rate to help overall connectivity, as the

lower the frame rate the easier it is to decode.

Though this is not as necessary in 2018, there are still territories with limited overall bandwidth, like Brazil, that use lower frame rates to compensate. When the choice has been made to use a lower frame rate, typically it is only done to the lower bitrate variants/flavors – around the 100 Kbps-600Kbps range. Variants/Flavors over 600 Kbps would get the normal, source frame rate up to 30 fps.

If a territory has to stream at a low frame rate it also means they cannot stream a high frame rate of 60 fps.Frame rates were determined initially based on the mid-20th century power grid and the Hoover Dam's 60Hz rate. Therefore, video frame rate became 30 fps – later changing to 29.97 to fine tune the signal and reduce artifacts. Film frame rate however was and still is 24 fps.

When digital video took over video production in the early aughts the frame rate of 23.98 was created (for a similar reason as the use of 29.97). 23.98 is what is most widely used in film and television streaming in 2018 though production frame rates are now being done as high as 60fps. Typical frame rates are:

- 15
- 23.98
- 24
- 25
- 29.97
- 30
- 60

VIDEO & PLAYBACK ARTIFACTS

If the typical frame rates settings are not followed, a variety of playback artifacts or errors may appear. These artifacts may include:

- Buffering – If there are huge bit rate spikes or if file sizes are large for various bit rates a player may stop playback and pause – loading the video and waiting for system resources to free up. This also may be because the buffer is set too low during transcoding.
- Frame Skips/Drops – Frames of video may drop if signal strength is not sufficient or if the wrong frame rate was used during transcoding. Resolution may also be a culprit of frame skips. If a user is using an older laptop over a wireless connection in their home, 720 and 1080 resolutions will most likely skip frames, creating a choppy playback experience.
- Stutter – Like frame skips, frames are dropping, but the perceived experience is that the video is stuttering. This may appear as if a frame is pausing for a split second before catching up to audio which normally does not stutter or skip. (audio typically plays back fine even if video artifacts are present as the two, though bundled in the same container, are decoded separately)
- Macro--Blocking (or just Blocking) – The image may appear as a mosaic of blocks – this is often referred to as pixilation. This especially happens on fast motion scenes or scenes with fine detail like rain. In the scene depicted here the lights behind the actor caused a number of lens flares that then resulted in blocking at lower bitrates. Macroblocking Example: Notice the squares/blocks rather than a smooth image.

- Aliasing – Lower resolutions do not handle diagonal lines well – these may appear as steps and be jagged rather than smooth diagonals. It can be especially noticeable on text.
  Aliasing Example: Notice the jagged steps on the diagonal of the 4 rather than a smooth line.

- Banding – Though on a video master an image may appear to have a consistent color, often after transcoding to mid to low bit rates/resolutions that same color appears as a gradient or a series of bands. This is referred to as banding. An encoder, especially if 1--Pass or CBR was used, may not be able to tell the difference between fine color changes and instead presents them as big changes, rather than subtle shifts. This is most often in sky shots and in animation; the latter is most problematic and challenging to transcode – especially modern CG based animation. Many transcoders have built in algorithms to deal with banding and over time this issue should dissipate.

AUDIO

Much of streaming video got it's start in the need to digitize audio for CD usage in the 1980's – in fact the initial bitrate limitations for video had everything to do with the 1.5 Mbps data rate of CDs and CDRs.

Just like Video, Audio requires a Codec and has specifications around bitrate and sample rate as well as track layout. And not all codecs are compatible with streaming as the player must be able to decode them. Uncompressed audio is not used for streaming video therefore some compression must be applied.

STREAMING AUDIO CODECS

AAC or Advanced Audio Codec, is the most common audio codec in use today in 2018. Its compression does a good

job of preserving source file signal to noise ratio and overall fidelity. (Google's WebM format uses a similar codec called Vorbis audio)

HE-AAC (High Efficiency-AAC) is a more advanced version, with better compression and dynamic range.

AUDIO – SAMPLE RATE

A sample rate is similar to a video resolution in that it governs the quality of the audio in how closely it matches the source audio. Think of each audio sample as a line of pixels of video, the more lines of resolution the sharper and clearer the image – the more samples per second the better the fidelity of the audio. The Sample Rate is measured in kHz for streaming video application. The standard audio sample rate for streaming video is 44.1 kHz or 44,100 samples per second. 44.1 kHz is the sample a rate for CD audio, DVD and Blu-Ray audio in addition to being the standard for streaming over the internet.

A source file may use the higher quality 48 kHz.

Regardless of the codec the bitrate and sample rates are common among the formats. So audio using AAC, HE-AAC and Vorbis Audio will all use 44.1 kHz for streaming audio.

AUDIO CHANNELS

Each stream of audio can contain multiple channels. Each channel may contain different audio from each other depending on the use case and formatting.

- **STEREO:** Normally we use 2-Channel Stereo audio – meaning there is a left channel, meant for a left speaker and a right channel, meant for a right speaker. Each of these channels typically contains the same information and is meant to create a balanced listening experience. These two channels will be embedded into 1 single audio stream and matrixed together. The player and playback device will then know to take part of the audio stream and send it to the Left Speaker and the other channel in the stream to the Right Speaker. Some sound engineers will take advantage of stereo's 2 channels by "panning" sound fx or music from one channel to the other – making it seem as if the audio is moving along with the picture (say panning the sound of a car passing through the frame left to right to match the left right movement of the on screen image).
- **MONO:** Older content and some UGC (user generated content) shot on SD video devices like older smart phones might use Mono, or single channel, audio. In the beginning of film sound only one speaker was used to present the sound track for a movie. This meant that all the audio would only be heard from the center of the screen – no fancy surround or stereo panning.
- **5.1 AUDIO:** Theatrical audio uses surround sound to immerse a user into a movie – putting some sound effects in rear speakers to make sounds move from the back of the audience to the front and vice versa for a desired effect. (say a plane flying from behind the audience towards the front of the audience as the plane appears on screen doing the same) There are many surround formats from 5.1 to 6.1 to 8.1 – for streaming video and home theatre usage typically we only deal with 5.1 audio. Each channel in a multi-channel audio mix represents a speaker in the listening environment of the theatre or home entertainment system.

  The 5 in 5.1 stands for the following 5 discreet channels of audio:

  -
    - Channel 1: Front Left
    - Channel 2: Front Right
    - Channel 3: Center
    - Channel 4: Rear Left Surround

- Channel 5: Rear Right Surround

  And the .1 stands for the LFE, or Low Frequency Effect channel, as the speaker for this channel of audio is only designed to carry low frequencies below 120 Hz.

AUDIO – SOURCE FILES

Source file audio can be very different than the resulting streaming variant audio. A source file may have multiple audio streams with multiple channels of audio, whereas a streaming variant/flavor will typically only have one 2-Channel Stereo stream. When multi-channel audio sources are used it may be necessary to downmix or otherwise map the audio so that the resulting transcodes are true 2-Channel Stereo. When a transcoded file is created it takes typically the first Audio Stream and the first 2 Channels of that Audio Stream and outputs a single 2-Channel Stereo Stream.

It can be easy to make a mistake and grab only the Front Left and Front Right Channels from a 5.1 audio source and create a 2-Channel output. The problem with this scenario is that the Front Left and Front Right channels typically do not contain Dialog in a true 5.1 mix. So, if you do use the Front Left and Front Right to create a stereo output it might contain little to no actual dialog – which is of course not a good thing.

It is vital to look at your source formatting when creating a transcoding strategy. There is not much consistency in the industry for source audio formatting. The layout of a source file may vary greatly from vendor to vendor or even from department to department within a single organization.

And 5.1 audio is not the only usage of multi-channel sources. A studio might create a video master that also contains several different language dubs that the film/content is meant to be heard in.

A single source file meant for North America might be formatted thusly:

- Stream #1: Video
- Stream #2: English 2-Channel Stereo
- Stream #3: Spanish Mono
- Stream #4: French 2-Channel Stereo

However, another source file might have this layout instead:

- Stream #1: Video
- Stream #2: Front Left
- Stream #3: Front Right
- Stream #4: Center
- Stream #5: Rear Left Surround
- Stream #6: Rear Right Surround
- Stream #7: Stereo Downmix
- Stream #8: Commentary

A Transcoding engine will need to be told how to map these different audio source types correctly to output a stream able file.

MULTI-AUDIO PLAYBACK

Modern streaming players can handle multi-audio on the fly switching, so it is now common to find multi-language audio sources where each stream needs to be pulled and transcoded into a standalone, audio only variant/flavor. The video

and audio will be re-synced once in the player. This means that an adaptive set for multi-audio content might include 6-8 video variants/flavors and an additional 2-8 audio only variants representing the different languages or audio types (commentary, visual impaired audio).

SUBTITLES

Subtitles are used now at the player level and are normally formatted in the SRT format though other formats also exist. The use of subtitles is often mandatory for any content that has been broadcast on national television in the United States to comply with standards around those who are hard of hearing or deaf. For broadcast television and DVD these subtitles are called Closed Captions and the file itself is stored inside the video signal on line 21 of the horizontal blanking (lines of resolution in a television).

For streaming video this Closed Caption file is often converted into an SRT file. There is a big difference however between a normal subtitle and a Close Caption for the Hearing Impaired even though they may initially appear to look very similar. Whereas normal subtitles only present the dialog as spoken (often with paraphrasing for timing), Closed Captions include sound descriptions as well and also character names.

A Subtitle might read: "**I told you I was leaving!**"

A Closed Caption for same scene might read: **[Chris] I told you I was leaving! [SLAMS DOOR]**

Subtitles, like audio, can be switched on the fly at the player level. You can also burn in subtitles into the video stream so that they always appear – this is sometimes used for downloadable video files or use cases where a player with subtitle switching is not available.

LANGUAGE CODES

Both audio and subtitles will use language codes so that a player can correctly label the audio track. This is done at the transcoding stage and is presented as a Label or Language Metadata field in the file. Though customizable, there are standards that should be followed for 3 letter abbreviations of languages. ISO 639-1 is the specification for proper language labelling.

You can find the ISO approved language codes here: https://www.loc.gov/standards/iso639- 2/php/code_list.php

STREAMING PROTOCOLS

Streaming video is built on the back of the internet's HTTP or Hypertext Transfer Protocol – the fundamental language of the internet where a client (a user's browser) makes requests from a server (a website like Facebook) for a variety of files that are then sent in streams of packets of data.

It should not be surprising then to find that two of the original protocols for streaming video are Adobe's HTTP Dynamic Streaming and Apple's HLS (HTTP Live Streaming). Unlike Progressive Download, both Live Streaming solutions allowed multi-bitrate switching (aka Adaptive Playback), better seeking, bandwidth control and of course actual live streaming of live events.

The way it works is, unlike Progressive Download solutions, the video file does not have to fully download first before playback, but rather each MP4 variant is segmented into smaller chunks. This is similar to the method used for Broadcast Television called MPEG-TS (Transport Stream) where broadcast television is sent in similar chunks or segments.

A manifest file is created that includes a list and the location on a CDN of each variant in the adaptive set and which variant should play first. It is wise to serve up a lower bitrate and resolution first in order to have fast start of video playback.

The basics of this methodology are still used in HTML5 Video and MPEG DASH.

HTML 5

HTML5 Video is the newest standard that replaced the widely used Adobe Flash. Unlike the old Adobe Flash Player, HTML5 does not require a user to download a plugin or keep anything up to date other than their browser. It is widely utilized and the spec for it allows for a wide variety of format compatibility. Virtually any video file can be played back using HTML5 Video - though that does not mean you should try to stream high bitrate source files as you still must follow the Adaptive Streaming rules of the digital road.

MPEG DASH

One of the primary methods for streaming video today is the use of MPEG DASH aka Dynamic Adaptive Streaming HTTP. It uses the same basic methodology of HLS and HTTP Live Streaming and even traditional MPEG-TS broadcast video in that it segments each variant into typically 2-10 second chunks so that the receiving player can handle the incoming bitstream without too much effort. It also allows a user to skip around from segment to segment much faster improving seek functionality. This is done often on the fly at the server level.

MPEG DASH also utilizes segmentation a manifest file that determines variant order (which variant loads into the player first).

DEVICES

As described in the Levels section in this article, not all settings apply to all devices. In 2018 there are a number of different devices and means by which video might be streamed. These range from the traditional laptop or desktop computer to smart phone or even to your home Smart TV. Video game systems also stream video today as well as Blu-Ray players and other connected devices. Bitrates, profile levels and resolution spreads all differ on these devices with some cross over. In order to reach all devices custom variants will need to be created with specific device tags.

Devices fall into the following 3 primary categories:

- Desktop
- Mobile
- Connected Devices

What is great about 2018 Streaming Video and HTML 5 is that the same variants, if you are thoughtful about how your format them, can be served up into multiple devices. In the early days of streaming a Desktop might be pulling VP6 (an antiquated streaming format) but an early Smart Phone might be streaming an MP4 file. Now both use cases can access the same MP4.

DESKTOP

Desktop refers to either a home tower type computer or a laptop that utilizes a browser to

access the internet over either a wired Broadband connection (Ethernet) or a WiFi network. The limitations of a desktop

environment extend to the computing power of the individual device given it's age and operating software. Older desktops and laptops will have slower speeds and capability compared to a newer system.

Video transcoding settings however are fairly flexible with desktop and laptops being able to handle all profile levels and resolutions up to 1080p. (4K is still limited in it's streaming usage as of this writing and most computers still struggle streaming full 4k.)

STREAMING VARIANTS *CONTAINER: MP4 /VIDEO: H264 CODEC / AUDIO: AAC CODEC @ 44.1 Khz* *FRAME RATE: Match Source up to 30 fps*			
VIDEO			AUDIO
RESOLUTION	BITRATE	PROFILE	BITRATE
480x270	200	Baseline @ L3	64
640x360	400	Baseline @ L3	64
640x360	600	Baseline @ L3	64
640x360	900	Main @ L3.1	64
1280 x 720	1500	Main @ L3.2	128
1280 x 720	2500	High @ L4	128
1920 x 1080	4000	High @ L4.2	128
1920 x 1080	6000	High @ L4.2	128
2560 x 1440	9000	High @ L5.0	128
3840 x 2160	13000	High @ L5.1	128

MOBILE

A smart phone or tablet in 2018 can stream 1080 content easily and even can record in 4K at 50 Mbps. However, there are several limitations and recommendations from device manufacturers to best use their devices.

GOP Size is recommended to be around 2-3 seconds, meaning a video at 30 fps would have a GOP size of 60 frames to 90 frames. As described, Key-Frames, when playing back, are considered IDR frames (Instantaneous Decoder Refresh) and are used as adaptive switch points and seek points for jumping around the video's timeline (skipping forward or back, etc.).

Resolutions and Profiles/Levels also must be carefully considered. The age of the device might mean it can only support SD variants or even only outdated formats like 3GP (used in early mobile phones with video).

Apple's HLS spec also calls for the use of an audio only flavor in case the users signal drops so low that streaming video is no longer possible – this assures the player keeps playing and can then adapt back up to a higher resolution that includes video.

ANDROID VP8 STREAMING VARIANTS CONTAINER: WEBM /VIDEO: VP8 FRAME RATE: 30 fps				
VIDEO			android phone/tablet	android TV
RESOLUTION	BITRATE (Kbps)	CODEC		
320x180	800	VP8	YES	NO
640x360	2000	VP8	YES	YES
1280 x 720	4000	VP8	YES*	YES
1920 x 1080	10000	VP8	YES*	YES

*On later devices only. Not available for all devices.

ANDROID H264 STREAMING VARIANTS CONTAINER: WEBM /VIDEO: H264 / AUDIO: AAC-LC FRAME RATE: 30 fps					
VIDEO				android phone/tablet	android TV
RESOLUTION	BITRATE (Kbps)	CODEC	AUDIO BITRATE		
176 x 144	50	H264	24	YES	NO
480 x 360	500	H264	128	YES	YES
1280 x 720	2000	H264	192	YES*	YES

*On later devices only. Not available for all devices.

iOS STREAMING VARIANTS CONTAINER: MP4 /VIDEO: H264 CODEC / AUDIO: AAC CODEC @ 44.1 Khz FRAME RATE: Match Source up to 30 fps / 16x9 Resolutions Shown				iPHONE			
VIDEO			AUDIO				
RESOLUTION	BITRATE (Kbps)	PROFILE	BITRATE	3	4	5, 6, 7, 8, X	ipad 2, Air
480x270	200	Baseline @ L3	64	YES	YES	YES	YES
640x360	400	Baseline @ L3	64	YES	YES	YES	YES
640x360	600	Baseline @ L3	64	YES	YES	YES	YES
640x360	900	Main @ L3.1	64	NO	YES	YES	YES
1280 x 720	1500	Main @ L3.2	128	NO	YES	YES	YES
1280 x 720	2500	High @ L4	128	NO	NO	YES	YES
1920 x 1080	4000	High @ L4.2	128	NO	NO	YES	YES
1920 x 1080	6000	High @ L4.2	128	NO	NO	YES	YES

CONNECTED DEVICES

Connected Devices are those that plug into your home television and also covers Smart TV's. This is a growing category of device that has, unfortunately, a wide variety of variables that differ from use case to use case. Not all devices are the same. The specs for an Apple TV are different than those for a ROKU and when newer models come out, the specs for each change, which means you might need to account for both older and newer models.

Other connected devices include Amazon Firestick, Chromecast and Gaming Consoles.

GAMING SYSTEMS

Xbox One, Xbox 360 and PS4 all can stream video. In fact the streaming on these devices can be superior to that of a desktop as each device is a dedicated video decoder and system meant primarily for video and audio playback.

However, each device has different settings that have to be closely aligned – especially when it comes to Profiles and Levels as well as frame rates. Some gaming consoles might require only Baseline profile whereas others might require all variants use High Profile.

Frame sizes may also differ – for example Xbox requires both full screen sizes and a smaller frame for when inset into their menus. Xbox also requires only High Profile whereas other systems require Main or Baseline only. Streaming to gaming consoles requires customization and finesse.

SMART TV's

Smart TV's are not as smart as other devices just yet. They are getting better and better but the built in memory of each television is still limited, causing apps to sometimes load slowly or other playback issues. Two primary manufacturer's dominate this market right now, Sony and Samsung Smart TVs who both use similar settings. Typically simpler profile levels are required.

SOURCE FORMATING

Transcoding always starts with a Video Master or Source File. The transcode can never be of a better quality than the Master.

"A transcode is only as good as the Master it was made from."

A master file intended for transcoding is often referred to as a Mezzanine file – as it's an intermediary step in the post production process. A true 4k uncompressed piece of video is far too large to even play back on most computers and certainly too big to transcode from as it would take a substantial amount of time to even move the files so that an encoder can pick it up and complete the transcode.

Ideally, the video master should be at the native resolution of active picture – in other words – if the video has an aspect ratio of 178 than the master should be 178, if the video, however, was 235 the master should also be 235 with no burned in matting.

Often Producers decide to use the master as is and allow the black mattes to be transcoded as well, rather than cropped out. This takes away from image quality as that black matte still gets compressed and still has bits assigned to it – therefore you are taking away resources from the actual active picture that may improve not only picture quality but the overall playback experience.

Now you can have a mezzanine or source file that is simpler and still get good quality. If the top variant/flavor is going to be, for example, 4000 Kbps, a source file that is 8000 Kbps would work just fine. There would of course be some quality differences from a larger source file at a higher bitrate but those would be negligible. Many news organizations, for example, utilize mezzanine files that are smaller bitrates as their use case focuses on speed to publish and they have a high volume of content being created every day. Therefore, using smaller source files helps them save storage space and time to publish.

The following is a chart detailing possible Master settings:

VIDEO				AUDIO		
CONTAINER	CODEC	RESOLUTION	BITRATE	CODEC	BITRATE	SAMPLE RATE
.MOV	ProRes/DNxHD	1920x1080	100 MBPS	PCM	1500	48 KHz
.MOV	ProRes	3840X2160	700 MBPS	PCM	1500	48 KHz
.MXF	AVC/H264	1920x1080	50-100 MBPS	PCM	1500	48 KHz
.MXF	AVC/H264	3840X2160	100-700 MBPS	PCM	1500	48 KHz
.MP4	AVC/H264	1920x1080	50-100 MBPS	AAC	1500	48 KHz
.MP4	AVC/H264	3840X2160	100-700 MBPS	AAC	1500	48 KHz
.MOV	H265	1920x1080	50-100 MBPS	HE-AAC	1500	48 KHz
.MOV	H265	3840X2160	100-700 MBPS	HE-AAC	1500	48 KHz

NOTE: NOTE: The GOP for any master should always be 1 frame, so that the file can easily be edited in a non-linear editing system.

THE MASTER STREAMING VARIANT CHART

Here is the full chart for H264 streaming variants/flavors. GOP/Key Frames should be the same for every level and should remain between 2-3 seconds.

STREAMING VARIANTS *CONTAINER: MP4 /VIDEO: H264 CODEC / AUDIO: AAC CODEC @ 44.1 Khz* *FRAME RATE: Match Source up to 30 fps / 16x9 Resolutions Shown*					iPHONE											
VIDEO			AUDIO													
RESOLUTION	BITRATE (Kbps)	PROFILE	BITRATE	WEB	3	4	5, 6, 7, 8, X	ipad 2, Air	android phone/tablet, Blackberry 10	android TV	Apple TV Gen 2	Apple TV Gen 3	ROKU	SMART TV	Xbox
480x270	200	Baseline @ L3	64	YES	YES	YES	YES	YES	NO	NO	YES	YES	NO	NO	NO
640x360	400	Baseline @ L3	64	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	NO	NO
640x360	600	Baseline @ L3	64	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO	NO	NO
640x360	900	Main @ L3.1	64	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO
1280 x 720	1500	Main @ L3.2	128	YES	NO	YES	YES	YES	YES	YES	YES	YES	YES	YES	NO
1280 x 720	2500	High @ L4	128	YES	NO	NO	YES	YES	NO	*YES**	YES	YES	YES	YES	YES
1920 x 1080	4000	High @ L4.2	128	YES	NO	NO	YES	YES	NO	NO	YES	YES	YES	YES	YES
1920 x 1080	6000	High @ L4.2	128	YES	NO	NO	YES	YES	NO	NO	YES	YES	YES	NO	YES
2560 x 1440	9000	High @ L5.0	128	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
3840 x 2160	13000	High @ L5.1	128	YES	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO

* Android devices recommend Main Profile @ L3 for 720 and above

References:

https://developer.apple.com/library/content/technotes/tn2224/_index.html#//apple_ref/doc/uid/DTS400097 45-CH1-ENCODEYOURVARIANTS

https://developer.android.com/guide/topics/media/media-formats?authuser=1

Additional Contributors:

Anatol Schwartz, Eran Etam, Eitan Lvovski