



Fundamentals of Multi-Channel Encoding for Streaming

This guide provides a simple high-level analysis of encoding and multi-channel encoding by breaking down the different technologies, approaches, and types of compression that can be used. It also highlights the benefits of each method by exploring the role multi-channel encoding plays in a variety of environments.

CONTENTS

- WHAT IS ENCODING?** 3
 - Transcoding and transrating 3
 - Encoding with or without compression 4
 - Real-time vs. non real-time encoding 4
 - Hardware vs. software encoding 5
 - Encoding for streaming and recording 6
- WHAT IS MULTI-CHANNEL ENCODING?** 6
 - Different ways to achieve multi-channel encoding 7
 - Benefits of multi-channel encoding 8
 - Change/augment protocols..... 8
 - Change/augment number of resolutions 9
 - Change/augment streaming profiles or video container formats 10
 - Deal with different network bandwidth in different ways and optimize for each case 11
- APPENDIX** 12

WHAT IS ENCODING?

Encoding refers to converting captured video and/or rendered PC graphics into a digital format that helps facilitate recording, moving, multiplying, sharing, altering, or otherwise manipulating the video content for editing, transport, and viewing. The process entails following a set of rules for digitizing the video that can be reversed by a “decoder”, to allow viewing. The decoder can be dedicated hardware or simply a software player. The encoding process can use a market standard or a proprietary encoding scheme.

First step: video capture

The first stage of encoding is video capture. This almost always involves capturing audio at the same time if available.

There are many different media that can be “captured”. Popular sources for video capture include: cameras, video production and switching equipment, and graphics rendered on PCs.

For cameras, video production, and switching equipment there are different ports to access the audio and video. Popular ports (I/O) from these devices that are connected to encoding equipment include: HDMI® and SDI.

Capturing rendered graphics or video from a PC can be accomplished in many ways. Software can be used to capture what is visible on the display of the PC. Another option is to capture the graphics output of the PC from popular ports such as DisplayPort™ or HDMI. It is even possible to do hardware-based capture from within the PC over the PCI-Express bus. Products

that support a very high-density of capture and/or encoding can be used in certain real-time recording or streaming applications of 360° video, virtual reality (VR), and augmented reality (AR), when combined with GPUs capable of handling video stitching from many IP or baseband cameras.

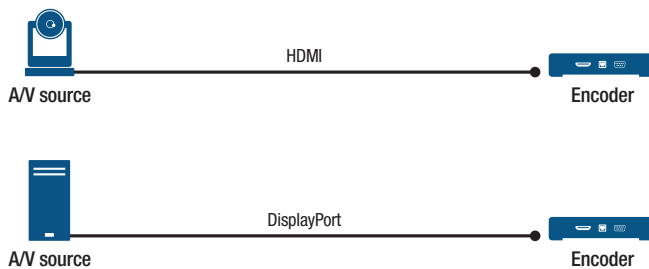
When using software encoding (see below), capture hardware for PCs comes in many forms including PCI-Express® cards, USB capture devices, and capture devices for other PC interconnect.

Next step: video encoding

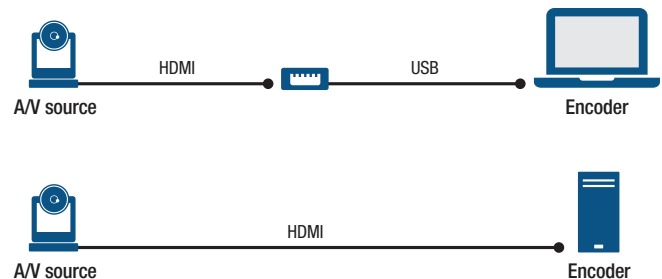
Encoding video can be achieved with hardware or software. There are features and price points in all granularities for the requirements of the workflow in both hardware and software.

There are many options for capturing and encoding video. Handheld mobile devices come with cameras and can create both encoded video files as well as live video streams.

Typical hardware encoder setups



Typical software encoder setups



Transcoding and transrating

Transcoding and **transrating** are other forms of encoding. This refers to taking digital video and converting it. An example of transcoding is taking a video asset from one format, such as MPEG-2, and converting it to another format, such as H.264. An example of transrating is taking a video asset and converting its resolution or bitrate characteristics but keeping the format the same; such as H.264 for example. For some operations of transcoding, video must be decoded and then re-encoded. For other types of transcoding, the same encoding format can be maintained but things such as the streaming protocols can be altered.

Sometimes software running on-premises or in the cloud as a service can be used for transcoding applications. The purpose and performance requirements of transcoding operations vary greatly. The amount of latency that can be tolerated by a streaming video workflow can impact choices for both the original encoding of various media and the transcoding options.

Encoding with or without compression

Encoding raw video can be achieved **with compression** and with **no compression**.

In video editing environments, for example, video is often manipulated, and many workflows are designed with digital uncompressed video.

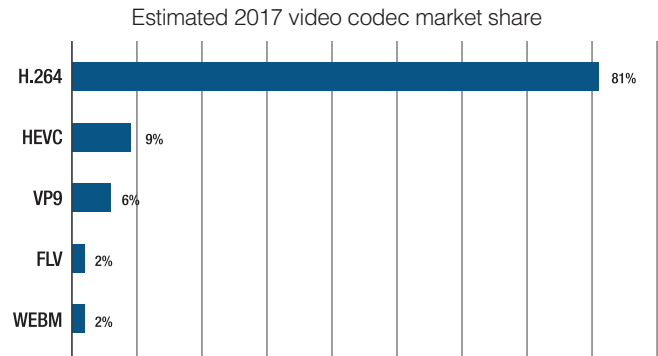
In applications where video is being served to users on the internet, video is usually compressed so it can fit on networks and be viewed on many different devices.

When video is made available directly from content owners to content viewers, by-passing cable and satellite service providers for example, this is sometimes referred to as “over-the-top” content or OTT for short. Almost all content that reaches a viewer, in any format, is compressed video. This includes OTT, Blu-ray, online streaming, and even cinema.

While video can be encoded (digitized) with or without compression, when compression is involved this usually involves a **video codec**, which is shorthand for: **compression/decompression**.

When the purpose of encoding is for live streaming or on-demand streaming of recorded media, video codecs—such as H.264—are used to compress the video. Software and hardware decoders reverse the process and allow you to view the media.

There are many on-line sources of information for evaluating popular codecs. Many reports still reflect a very dominant market share by H.264.



Real-time vs. non real-time encoding

Encoding video is an operation that can happen in real-time or something that can happen with more considerable latency.

Much of the on-line video available in streaming services for movies and shows, for example, uses **multi-pass encoding** to exploit compression technologies that offer viewers the best blend of performance and quality-of-service. Image quality and bitrate are normally inversely correlated where optimizing one penalizes the other. But the bitrates of video can be significantly mitigated using multi-pass techniques, while still producing exceptional quality and performance to viewers.

One of the major distinguishing features between hardware-based and software-based real-time encoders for applications over bandwidth constrained networks is the latency, quality, and bitrate optimization that they can achieve. The best encoders, both hardware and software based, can produce exceptional quality at very low latency and very low bitrates.

In other instances, **real-time video encoding** better suits the application. For example, in live streaming applications, where only very nominal latency is tolerable between the camera and the viewing audience, the video is often captured, encoded, and packaged for distribution with very little delay.

On-line meetings and web conferences normally use real-time video encoding as do professionally-produced live webcasts.

Note: the “on-demand” version of web conferences and webcasts that are recorded for later consumption by viewers on their own time are usually in the same format as the original live event handled by a real-time video encoder. This is because quality cannot be added back once the video goes through its original encoding with compression.

One of the major distinguishing features between hardware-based and software-based real-time encoders for applications over **bandwidth constrained networks** is the latency, quality, and bitrate optimization that they can achieve. The best encoders, both hardware and software based, can produce exceptional quality at very low latency and very low bitrates.

Sometimes encoders can also be tightly-coupled with corresponding decoders. This means that vendors offer both ends with certain additional optimizations. For example, the ease and automation to connect source and destination end-points, the signal management and switching, and the overall performance and quality can be tuned to supplement and augment or, in some cases, entirely replace traditional hardwired AV infrastructures.

Hardware vs. software encoding

The difference between hardware and software encoding is that hardware encoding uses purpose-built processing for encoding, whereas software encoding relies on general-purpose processing for encoding.

When encoding is performed by dedicated hardware, the hardware is designed to carry out the encoding rules automatically. Good hardware design allows for higher quality video and low power consumption and extremely low latencies, and can be combined with other features. These are usually installed in situations where there is a need for live encoding.

Software encoding also uses hardware but uses more general-purpose processing such as CPUs in personal computers or handheld devices. In most cases software encoding exhibits much higher latency and power requirements. The impact to latency and power using software encoding is even higher for high-quality video. Many modern CPUs and GPUs incorporate some level of hardware acceleration for encoding. Some are I/O limited and mainly used for transcoding. Others incorporate a hardware encode for a single stream, for example to share a video game being played.

A good example of a use for software encoding using high-quality video is video editing, where content editors save changes often. Uncompressed encoded video is used to maintain quality. At the end of the video editing process, re-encoding (transcoding) the video, this time using compression, allows the video to be shared for viewing or stored in a reduced file size. While uncompressed video usually remains stored somewhere for future editing options, extra copies of the video, used for viewing, are often in compressed format. Moving uncompressed video is extremely heavy on bandwidth. Even with new high-bandwidth networks, effective bandwidth and scalability are always maximized when video is compressed.

Another example of software encoding can be using a personal computer's camera or a smart handheld device to carry out video conferencing (or video calls). This is often an application of highly compressed video encoding carried out in software running on CPUs.

To users, the distinction between hardware-accelerated encoding versus software encoding can be nebulous. Hardware acceleration serves multiple different purposes for different workflows. For example: many handheld devices contain CPUs that can accelerate the encoding of highly compressed video for applications such as video calls. The “goal” of hardware acceleration in this case is to protect the battery life of the handheld device from a software process running on the CPU of said device without acceleration. Left to run entirely in software, video calls, watching streaming video on YouTube, or watching videos stored on the phone, would all be activities that would significantly drain the battery life.

There is a correlation between the “complexity” of the encoding task with respect to whether software encoding—running on general-purpose computing—is used, or whether hardware-accelerated encoding is used. Maintaining video quality while significantly compressing the size of the video for storage or transmission on networks is an example of complexity.

This is one of the reasons why video standards are very important. The fact that H.264 has been a long-serving video standard has meant that it is hardware-accelerated in smart handheld devices and personal computers. This has been one of the major reasons it has been so easy to produce, share, and consume video content.

Streaming video services offering home users with movies and shows sometimes use software-based encoding to achieve the highest quality at the lowest bitrates for reliable high-quality experiences to millions of concurrent users. But for such a targeted use-case, they use a large number of computers for very long runtimes to find the most optimal encoding parameters. This is not done in real-time and is more suitable for on-demand streaming, versus live streaming.

Video quality is important. Maintaining quality while fitting on networks for reliability and performance is critical. Keeping encoding latency low, video quality high, and bandwidth low is essential for live streaming applications.

For encoding applications with more narrowcast applications, such as video editing infrastructures, it makes sense to use less complex processing for uncompressed or lightly-compressed encoding.

For corporate, government, education, and other organizations that produce a lot of video for their own consumption (versus video that is produced for sale to consumers), there is a need to balance many variables. Video quality is important. Maintaining quality while fitting on networks for reliability and performance is critical. Keeping encoding latency low, video quality high, and bandwidth low is essential for live streaming applications. “Recording” for on-demand streaming is often performed in the same step as encoding for live streaming. So the high-bandwidth approach of video editing infrastructures is not practical here. And the highly-optimized multi-pass encoding approach from movie streaming services is both out-of-budget as well as non-real-time and does not fit many applications from these organizations.

Encoding for streaming and recording

Encoding the video is only the first step in the process for streaming or recording. So how does the encoded video get from the encoder to the viewer, or to the recording device? The encoder needs to send the video somewhere, but it also needs to tell the receiver what it is sending.

Streaming protocols are different video streaming delivery rules and optimizations that are encapsulated to deal with different objectives and priorities such as video latency, network bandwidth, broad device compatibility, video frame rate and performance, and more.

Streaming protocols allow video that has been encoded to subsequently be transported, either in real-time or at a later time. Protocols do not affect the video itself, but rather how a user/viewer might interact with the video, the reliability of delivery of that video stream, or which devices/software players can access it. Some protocols are proprietary and can only be used by specific vendor hardware, significantly reducing the interoperability and potential reach of that content.

Simplistic AV-over-IP products in the AV industry often produce these proprietary stream formats that increase vendor lock-in, reduce interoperability, and greatly reduce flexibility for how the assets can be used in organizations. But they take responsibility for the interoperability of their own products. Sometimes customers willingly accept this lock-in to increase their confidence that large groups of distributed end-points will seamlessly work together and that vendor support will be clear in the case of incompatibilities, bugs, or other problems.

Different protocols are designed for different applications. For example, on a local network when sharing a live event, latency will be a key component. The viewer will not necessarily need playback controls and network reliability can be assured by some organizations, so there may be less of a need to employ sophisticated error correction. So protocols that are used across cloud or public internet may be different than protocols used for facilities AV infrastructure over IP.

When diffusing a stream to multiple platforms for wider distribution on the internet, HLS, MPEG-DASH, and Web RTC are among the protocols used to distribute content broadly. Prior to using these protocols for stream diffusion, the stream protocols used for uploading content from a facility to cloud services might be things such as RTMP. Where networks are unreliable but video quality still needs to be maintained, or the video needs to be secured, newly emerging protocols, such as SRT, might also be entirely appropriate.

Secure Reliable Transport (SRT) is a new protocol that was developed as a replacement candidate for RTMP. Many hardware and software companies have already implemented support for this new transport protocol.

There are many more protocols than mentioned here, each offering different strengths.

When the video is being stored, rather than viewed as a live stream, it requires a method of storage. Unsurprisingly, there is a wide gamut of options here as well for storing uncompressed, lightly compressed, and highly-compressed video. While operations can be performed on stored video to make it consumable with different options at a later time, the more thinking goes into how the stored video will ultimately be consumed, the more decisions can be made up-front about how to digitize it at the capture point. Just as in the streaming discussion above, there are tools for every workflow. And in the context of this multi-channel encoding discussion, many options for storage can be dealt with directly at the capture point and/or with transcoding using media servers and other tools.

WHAT IS MULTI-CHANNEL ENCODING?

Multi-channel encoding refers to the ability to serve multiple simultaneous streams from captured video sources. This is most useful for making a media source available to many destinations for immediate consumption (live streaming) and later consumption (on-demand streaming). Multi-channel encoding deals with problems such as: number of simultaneous viewers, types of viewing options (hardware vs. software, wireless devices, etc.), and recording options for on-demand streaming at a later time.

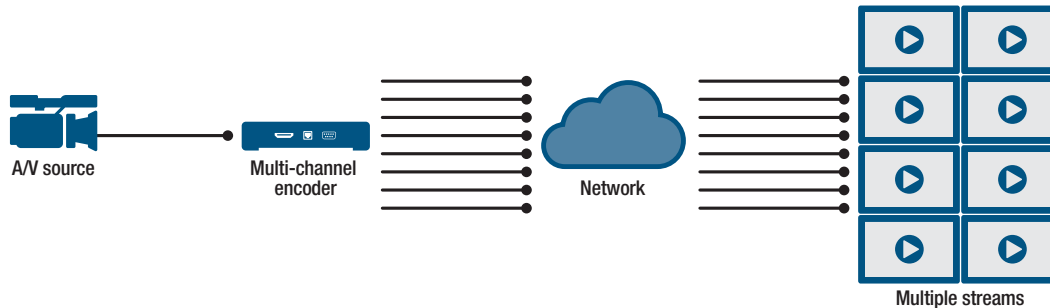
While video production environment workflows often deal in uncompressed video to maintain quality throughout the editing process, **most applications of multi-channel encoding deal with compressed video** for facilities AV and for content distribution across multiple locations and through the public internet.

Different ways to achieve multi-channel encoding

There are multiple different workflows for creating multiple streams.

Using a multi-channel encoder

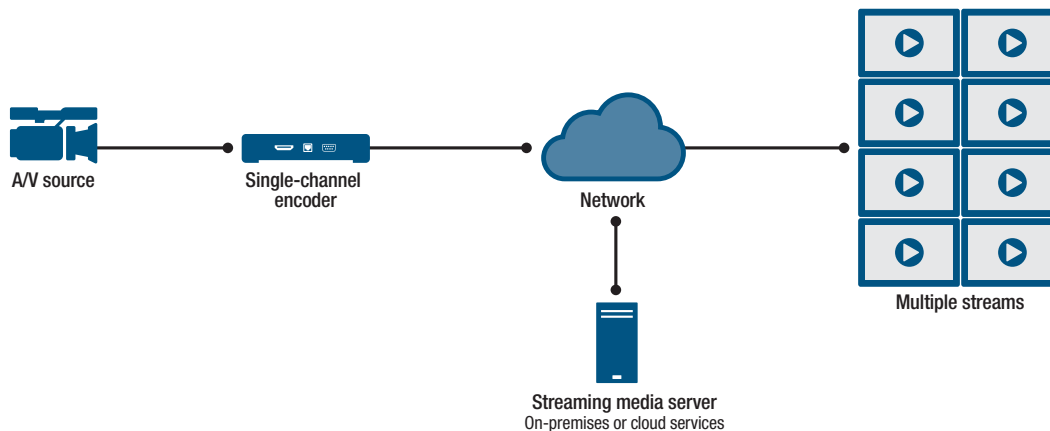
One way to generate multiple different streams is by using encoders that have the processing power and features to produce multiple streams directly from the encoder.



The benefit of using a multi-channel encoder is that less hardware is required further down the pipeline. Configuration of the desired channels can be performed and tested locally. This type of encoder will often be more sophisticated, with more features and flexibility than cheaper encoders, and is often capable of higher quality video as well.

Using a streaming media server

Another way is to use **streaming media servers**, which usually means software running on dedicated appliances, PCs, or servers, that basically takes source streams as inputs and uses the processing power of the streaming media server to transcode and multiply the number of available streams. Some streaming media servers run on-premises. Some streaming media servers run in the cloud.



There are many types of media servers. Some are for serving media content at home. Some are for performing transcoding operations for enterprise video distribution. Media servers are very useful to enhance the functionality of any type of encoder. However, they either require additional hardware (for on-premises media servers) or subscription to a service provider (for cloud-based servers), and sometimes both.

While streaming media servers offer flexibility (especially for cloud-based services), they cannot improve the quality of the video that they receive. As such, if option A is to use a high-quality, multi-channel encoder streaming direct, and option B is to use a low-quality single-channel encoder in conjunction with a streaming media service the cost might come out to a similar level, or even slightly cheaper for option B, but the distributed video in option A is going to be far superior.

But streaming media servers and multi-channel encoders are not mutually exclusive. For example, you might use a multi-channel encoder to provide multiple resolutions on a local network at an event, and use an additional channel from that multi-channel encoder to send a stream off-site to a streaming media server. Alternatively the multi-channel encoder can send one stream to a network attached storage (NAS) device and a second stream to a streaming media server. In both cases the multi-channel encoder is capable of meeting the local requirements **and** sending a high-quality video to the streaming media server for mass distribution.

Benefits of multi-channel encoding

Using multi-channel encoders and/or streaming media servers provides multiple advantages.

1. [Change/augment protocols](#)
2. [Change/augment number of resolutions](#)
3. [Change/augment streaming profiles or video container formats \(file formats\)](#)
4. [Deal with different network bandwidth in different ways and optimize for each case](#)

Change/augment protocols

Since different video streaming protocols deal with different problems, it makes sense that **multiple different protocols** are sometimes required to get video from media sources, like cameras, all the way to many simultaneous consumption nodes like smartphones, tablets, PCs, media players, and game consoles, and over very large distances to a disparate base of viewers. This often necessitates the use of cloud services or the public internet.

Having multiple protocols and the ability to change protocols for different segments of your workflow allows you to maximize both reach and performance

For example: “continuous” streaming protocols, like RTMP, can help maintain certain aspects of video performance while minimizing latency.

HTTP-based protocols, like HLS and MPEG-DASH, package video streams into fragments to better borrow the massive interoperability of networks and software applications by behaving like all other network traffic. They rely on TCP transmission to provide error correction, and on HTTP to traverse firewalls without requiring special instructions. However, these protocols require huge amounts of buffering to make this all work which injects significant latency. These solutions are perfectly acceptable for on-demand streaming workflows. But the market is working very hard to continue to compress latency for live streaming applications.

So having **multiple protocols** and the ability to **change protocols** for different segments of your workflow allows you to maximize both reach and performance by providing you with the ability to have some more advanced nodes capable of maintaining a low latency and very high video performance while also assuring that everything else is “compatible” with your streaming delivery set-up.

This applies at a local level just the same as it does over the internet.

Local

At a local level, an encoder running on a decent network can feed directly into a decoder and provide high-resolution video with minimal latency. If it is a multi-channel encoder, the same encoder can provide **additional** streams that work with standard players and browsers on lower bandwidth parts of the network, including wireless devices. Whether or not your encoder supports multi-channel encoding, it is also possible to use a streaming media server on your network to multiply the streams and/or change the protocols to suit your applications.

Some manufacturers of encoders also provide hardware and/or software decoders—minimizing complexity to have everything work together seamlessly.

“Recording” for on-demand streaming can also be fairly mission-critical in order to avoid losing a keynote speech or important moment during a network interruption. Sometimes multi-channel encoders and/or encoder and streaming media server combinations provide a local cache of what’s being recorded while simultaneously recording on cloud services. Or recording and simultaneously live streaming captured video sources may be the desired application. Here too, different protocols may be called into service such as FTP for an MPEG-4 file recording and a live RTMP H.264 stream.



Cloud/internet

The same applies to cloud/internet where-by multi-channel encoding enables the use of the right protocols for the right segments of the video streaming workflow.

By leveraging the appropriate protocols it is possible to have a mix of very high-performance nodes and very easy-to-access nodes. Protocol flexibility also allows you to mix old/legacy compute equipment with much more modern equipment. This means it is possible to pursue continuous improvement and evolution of your video streaming infrastructure instead of requiring large overhauls and revolution of your infrastructure.

Many cloud streaming architectures currently use a low-latency protocol, such as RTMP from the video source to the cloud and use more broadly compatible HTTP-based protocols for mass distribution.

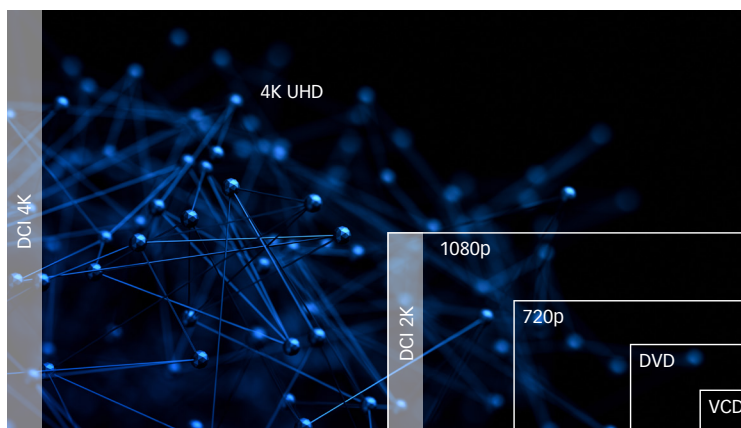
The multi-channel load can be placed on the encoder or the streaming media servers being used or a combination of the two.

Recorded files

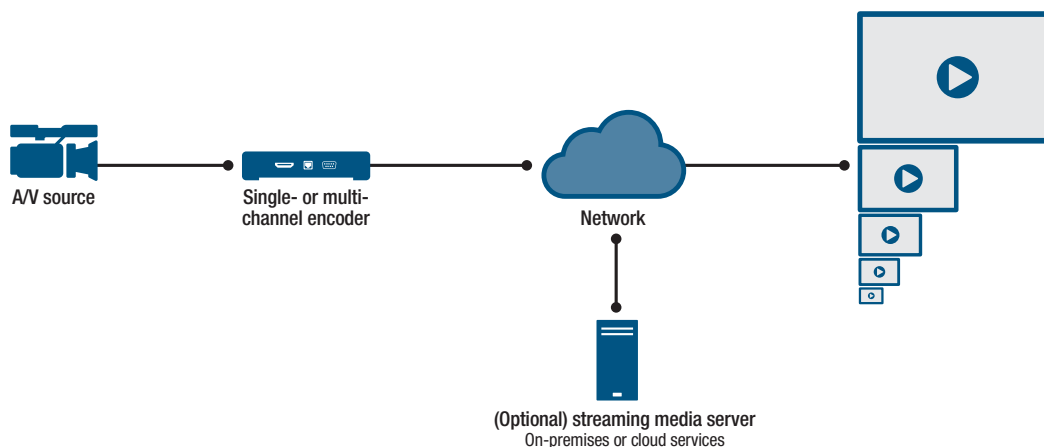
Another instance of a change in protocol is when streaming from a stored file rather than a live source. A perfect example of this is Video on Demand (VOD) services. These providers must store the video content in a container, and when a user initiates a viewing session it then converts it from a stored file to a video stream which is sent to the viewer over the internet. This could be handled by a multi-channel encoder or streaming media server. The protocol that they will use to communicate with the viewing device (such as a SmartTV) will help inform them of the bandwidth availability and reliability of the network, which allows them to select the appropriate resolution stream to create/send from the stored file.

Change/augment number of resolutions

One of the most important variables that affects the bitrate of live streams is the resolution of the video being streamed.



Multi-channel encoding deals with this problem as well.



Delivering streaming video is a balancing act between visual acuity and stability of the stream. In the early days of watching videos from the internet, users often experienced the frustration of buffering. Many videos were simply un-watchable.

Significant progress has been made to deliver optimal experiences that account for how much bandwidth is available and how much information can be carried in the video streaming payload to each node. (Higher resolutions require more information.)

Today, adaptive bitrate streaming technology automatically detects users' bandwidth and computer processing availability in real time and provides a media stream that fits within these constraints.

Transcoding in the cloud is something that creates latency and requires paid-for services. For this reason, many organizations that generate a lot of private (corporate) video content are balancing the load by either sending multiple different resolutions from multi-channel encoders from each captured video source in their organization, or using adaptive bitrate encoders that can be leveraged by certain compatible multimedia players that have the ability to switch between the different bitrate segments and offer the maximum quality (often includes resolution) that optimally suits the compute power and network conditions of that player node.

In enterprise and media and entertainment encoding, this basically means that video sources are often sent at their maximum quality and resolution profile but the local encoder and/or streaming server also create additional stream copies of the source in reduced settings.

This “scaling” of video sources through multiplication of the streaming video profiles is very useful to instantly accommodate all destination types. A 4K source, for example, can be kept in 4K and decoded at an appropriately powered viewing node. But the same 4K source can comfortably supply the same source content onto tablets and smartphones. These devices often have a lower resolution screen anyway and the corresponding reduced resolution stream is served to match what the wireless network and processing power of these wireless devices can handle.

Change/augment streaming profiles or video container formats

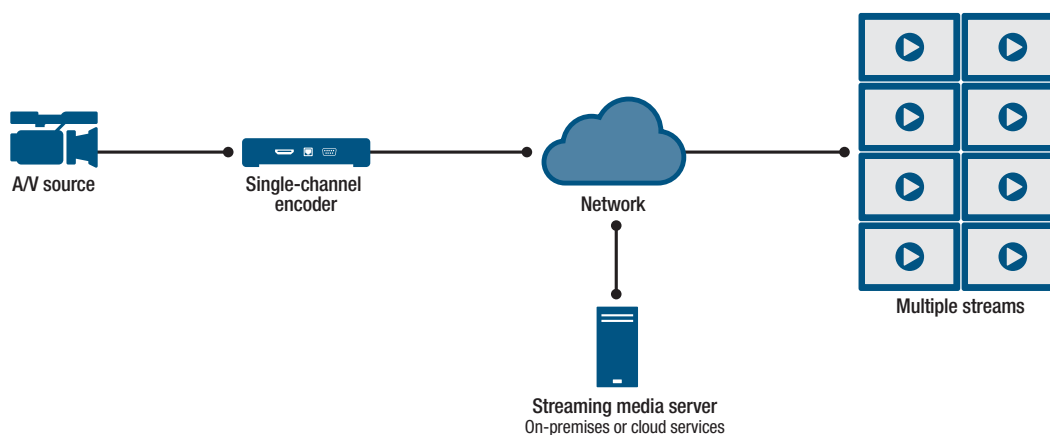
One of the most important variables that affects the bitrate of live streams is the resolution of the video being streamed.

Another aspect of multi-channel encoding is the ability to convert assets from one streaming codec or video file format to another or to multiple others. This can be more processing intensive than changing protocols as in the example above. Going from one codec to another often requires decoding the original stream or file and transcoding it (re-encoding it) to one or more different codecs or file formats.

There are different motivations for changing the codec of your video assets.

Here is a simple example:

Assume an organization has added new equipment capable of generating very high resolution, such as 4K. When these new assets are captured at full resolution, using codecs that produce a small-enough bandwidth might be enticing. But the codec and/or encoding profile used directly from the source to mitigate its bandwidth use may not match what is the optimal codec or encoding profile for content distribution at large.



Using HEVC (H.265) to encode 4K content may appear to shave off some bandwidth and help assure the stability of the stream from its capture point to its stream re-distribution point on a network or on the internet. But HEVC tends to drain battery on handhelds more than H.264 and many older devices do not have hardware implementations of HEVC. Media servers and other tools are therefore still extensively used to turn new HEVC sources into more convenient H.264 streams for many applications.

Conversely, some installations have legacy MPEG-2 sources. In this case, a transcoding effort could mitigate distribution bandwidth 'and' augment downstream device compatibility.

Not All Encoders Are Created Equal

It should be noted that there is a big gap in performance between encoders. Some highly-optimized H.264 encoders can produce bitrates that are superior to some early or basic HEVC encoders. The same applies for other encoding performance metrics such as latency or image quality.

But over time there are transitions in the market for resolutions and codecs. At some inflection points it sometimes makes sense to use different technology from the source-side encoder to the content delivery infrastructure versus the content delivery network to the final consumption nodes. Archiving the highest resolution content is sometimes a good enough excuse to move to less established technologies to mitigate storage costs. But mass distribution always requires well-established technologies for maximum compatibility and reach.

Transcoding can be expensive. It makes sense to study what can be achieved to minimize transcoding burdens on a video distribution infrastructure. When a video library is archived in a

highly compatible format it may still be the better compromise to use a well-established codec, like H.264, right from the get-go. Some emerging standards falter or get skipped. And some well-established standards continue to generate more evolved implementations and have a very compelling mix of performance and broad compatibility.

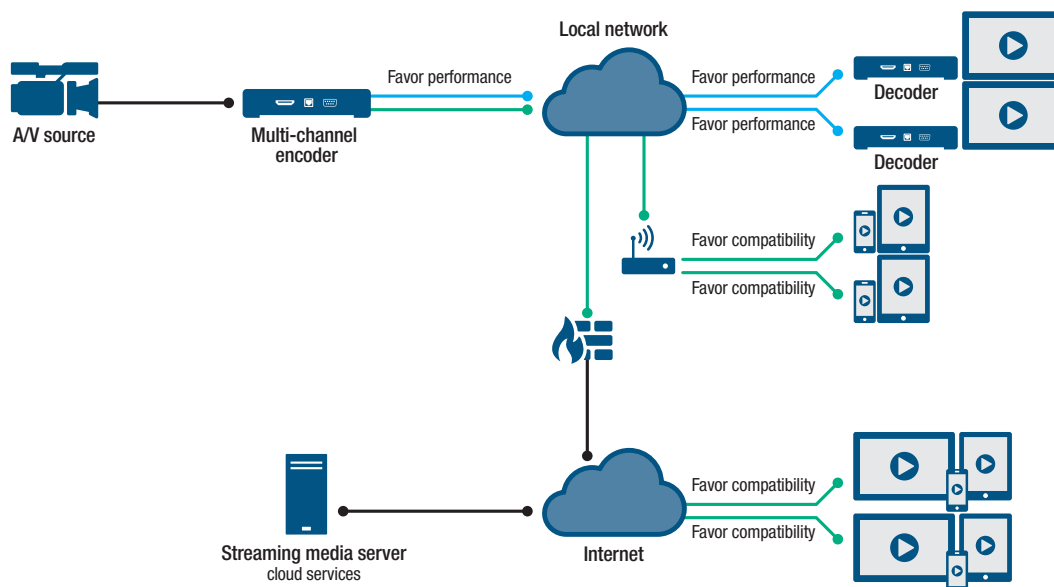
But whatever your workflow requires, multi-channel encoders and transcoding software and services can often assist with moving between codecs and encoding profiles and helping you reach your viewers.

Deal with different network bandwidth in different ways and optimize for each case

All three previous sections above combine to demonstrate how supporting multiple protocols simultaneously, how transcoding and transrating, and how producing different resolution and quality streams to deal with different bitrates and decoder/players justify multi-channel encoding.

We also reviewed different methods of multi-channel encoding including: multi-channel encoders that produce multiple streams right at the source, adaptive bitrate encoders which produce multiple profiles for compatible destinations to choose from, and transcoding media servers—which are software and services that let you manipulate and multiply your source video streams to suit your application.

Hybrid environments that fully-leverage one or more of these multi-channel encoding technologies allow organizations to serve streaming content in the best ways to all points factoring in considerations of security, network bandwidth, number and type of decoders/players, and more.



Appendix

More information on the concepts discussed in this technical guide can be found below:

Adaptive bitrate streaming:

https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming

AV Streaming Protocols: What is Typically Used, Where, and Why (Webinar):

http://www.matrox.com/graphics/en/press/events/webinar/video_over_ip/

Buyers' Guide to Media Servers 2017:

<http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=117457&PageNum=1>

Comparison of video container formats:

https://en.wikipedia.org/wiki/Comparison_of_video_container_formats

Fundamentals of AV over IP technical guide:

<http://www.matrox.com/graphics/en/press/guides/av-over-ip-fundamentals/>

HLS and H.264 Top Dogs in 2018 Encoding.com Global Format Report:

<http://www.streamingmedia.com/Articles/News/Online-Video-News/HLS-and-H.264-Top-Dogs-in-2018-Encoding.com-Global-Format-Report-124315.aspx>

Multi-pass encoding:

<https://www.afterdawn.com/glossary/term.cfm/multipass>

Secure Reliable Transport (SRT):

<http://www.matrox.com/graphics/en/av-over-ip/srt/>

Streaming Protocols 101 (Webinar):

<http://www.matrox.com/graphics/en/press/events/webinar/streaming-protocols-101/>

Contact Matrox

video@matrox.com

North America Corporate Headquarters: 1-800-361-4903 or 514-822-6364

Serving: Canada, United States, Latin America, Asia, Asia-Pacific, and Oceania

London Office: +44 (1895) 827300

Serving: United Kingdom, Ireland, Benelux, France, Spain, Portugal, Middle East, Africa

Munich Office: +49 89 62170-444

Serving: Germany, Austria, Switzerland, Denmark, Finland, Norway, Sweden,
Central and Eastern Europe, the Baltic States, Greece, Turkey, Italy

About Matrox Video

Matrox Video is a global manufacturer of reliable, high-quality ASICs, boards, appliances, and software. Backed by in-house design expertise and dedicated customer support, Matrox products deliver stellar capture, extension, distribution, and display. Engineering high-quality products since 1976, Matrox technology is trusted by professionals and partners worldwide. Matrox is a privately held company headquartered in Montreal, Canada. For more information, visit www.matrox.com/video.