Streaming Video Fundamentals

By Jason Thibeault

Abstract

This article provides an overview of video streaming technologies, the workflow that delivers videos to end devices, and the new video experience streaming enables. This article assumes a basic understanding and familiarity with traditional broadcast production workflows, hypertext transfer protocol (HTTP), video compression codecs, media security (such as digital rights management), video advertising, software-defined resources, and content delivery networks (CDNs).

Keywords

Internet protocol (IP) video, over-the-top (OTT), streaming video

Introduction

This article explores the current state of video streaming technologies, the technologies that deliver video content to end devices, and how streaming technologies provide new experiences. We assume the reader is familiar with the following concepts:

- *Public cloud and infrastructure as a service:* large-scale service providers, such as Amazon Web Services (AWS) and Microsoft Azure, which provide network computation, delivery, and storage services on an as-needed basis to build virtual processes;
- *Caching:* the temporary storage of data, often near the end users, in a specialized server (reverse proxy) that handles initial incoming requests and is often used to store very popular content;
- *Media security:* a variety of technologies, ranging from simple data encryption to more complex and proprietary solutions, such as digital rights management (DRM) and watermarking, which can protect media assets from being viewed without proper permissions;
- *IP-based data transmission:* delivery of data using the Internet Protocol (IP). IP often uses the Transmission Control Protocol (TCP) as a host-to-host data transport protocol, although sometimes IP utilizes the User Datagram Protocol (UDP) along with higher level transport protocols (such as QUIC) to provide even more reliability.
- *Video codecs:* these methods [such as Motion Picture Experts Group (MPEG), MPEG-4, ProRes, and Windows Media Video (WMV)] provide a means of encoding (compression) and decoding (decompression) videos so that they can be viewed.

Ultimately, the shift to IP benefits everyone. For the consumer, it's a new content experience founded on choice—more channels become available as content owners establish one-to-one relationships, and the consumer can pick and choose what they want (rather than being forced into a single electronic programming guide (EPG) by a MVPD or cable operator). For the broadcaster, this transition enables them to establish direct relationships with their viewers as they deliver content directly, rather than through a distributor. Enter streaming.

Now that we have that out of the way, let us discuss streaming video in detail.

The Shift to Internet Protocol

To understand the basics of streaming, we need to understand why. Ultimately, shifting to IP-based video delivery is all about flexibility. When video can be transmitted via IP, it's no longer shackled to proprietary distribution equipment (like a set-top box), unlocking many new opportunities. For example, delivering video over the internet, allows for the transmission of content directly to consumers, enabling a more individualized content experience and a deeper two-way, more personalized relationship between the content producer or distributor and the consumer. IP also enables content owners and distributors to deliver video content to other endpoints such as PCs, connected devices, web browsers, and mobile phones directly and even provide for interactivity with the content. Ultimately, the shift to IP benefits everyone. For the consumer, it's a new content experience founded on choice—more channels become available as content owners establish one-to-one relationships, and the consumer can pick and choose what they want (rather than being forced

Digital Object Identifier 10.5594/JMI.2020.2976257
Date of publication: 15 April 2020

into a single electronic programming guide (EPG) by a MVPD or cable operator). For the broadcaster, this transition enables them to establish direct relationships with their viewers as they deliver content directly, rather than through a distributor. Enter streaming.

Streaming Is Not Broadcasting

Broadcasting a video signal for delivery to a television set is a very proven technology, governed by an incredible number of standards and laws that determine most of the delivery mechanism such as video resolution, frame rate, carriage of Closed Captioning, audio loudness requirements, and digital RF modulation techniques. Streaming, on the other hand, has been developed in a more "organic" and unconstrained fashion. It has one, underlying requirement: adherence to internet protocols. There have been a variety of methods used to stream video over IP such as Realtime Messaging Protocol (RTMP), Realtime Streaming Protocol (RTSP), and more. Each of these methods has had their pros and cons. They could deliver streams in realtime. But they were proprietary, requiring access through corporate firewalls (for a non-standard port) and plugins, like Adobe Flash or QuickTime, for web browsers. What's more, securing these proprietary protocols was difficult.

Streaming video via the Hypertext Transport Protocol (HTTP), on the other hand, takes advantage of the underlying technology of the World Wide Web. All web servers in the world deliver media via HTTP and ensure no extra software is required to be installed while sending videos. This has allowed video content to proliferate well past the traditional STBs and to a variety of endpoints such as mobile phones, connected devices (like a Roku or AppleTV), and even smart TVs. Consumption of streaming video has continued to increase because of the ease of availability from a variety of devices. According to Nielsen's 2019 Total Audience Report, about 19% of all video viewing in the United States (Q4) was through streaming (a 10% increase from Q1, 2018). Consumers can watch streaming videos, whether it is a live sporting event or on-demand from a video library, irrespective of wherever they are and by whatever network connection that supports the transmission of data over IP.

How Does Streaming Work?

When streaming uses HTTP as the delivery mechanism, the mechanism is very simple (although it is a little more complicated, we are going to radically simplify it):

- A user makes a request for a video file. This could be by either clicking on a "play" icon in a video player in a web browser or selecting an episode in a library on a smart TV interface or an app on the phone.
- The request is routed, by domain name system (DNS), to a server that should have the content. This server, which is a web server, can be an edge cache in a content delivery network (CDN) or housed in the content owner's data center (this is called the content origin). If the content is present in the server, the server returns the data, as a response to the request, and the video starts playing (after the video buffer fills). If the content is not available, a variety of things can happen. A CDN can search parent caches or even go back to the origin. Or the server could respond with a 404, meaning the content is not available anymore.
- Because a connection has been opened, it can be maintained by the web server with the requesting client (like a web browser). This repeats step 2 until the client stops requesting because the user pressed "stop" or the video is finished.

The Streaming Video Workflow

A typical streaming video workflow is comprised of the following stages:

- Acquisition
- Transcoding
- Manipulation
- Delivery
- Playback
- Analytics

Acquisition

This is the primary handoff from a traditional broadcast workflow. The content is either acquired directly from the encoder (usually this is an RTMP push but many encoders now support HTTP pull) or picked up from the storage (for on-demand delivery).

> **Can you "stream" a video file that is not a live event?**
>
> Once termed "pseudostreaming" (and now more often referred to as *segmented video streaming*), this is what any major on-demand over-the-top (OTT) platform, such as Netflix or Disney+ does. The file is never downloaded, instead provided as a stream of video segments that are not saved on the client device. Of course, this requires an active internet connection through a mobile or wireless network. Most of the OTT platform providers now also offer "download-to-go" features that enable users to download the entire video to their device for offline consumption. In such cases, the video is not streamed at all, but played locally.

Transcoding

Transcoding is the stage at which content is transformed into multiple qualities for adaptive bitrate (ABR) delivery. The result of these different quality versions is the "ABR ladder," which is used by the player to determine what bitrate is most appropriate based on analysis of the viewer's environment (available memory, available compute, available bandwidth, etc.). Note that most players, regardless of the screen resolution, will attempt to acquire the highest bitrate they can. Most of the streaming videos today, to be delivered via HTTP are transcoded using MPEG-4 and produce an .MP4 or .TS file. The specific implementation of MPEG-4 is through a codec like H.264. There are newer codecs, such as H.265, AOmedia Video 1 (AV1), and versatile video coding (VVC), which promise significantly improved compression and computational efficiency, thereby allowing more videos to be encoded faster at higher bitrates with lower storage requirements. But the codec used for encoding has an impact on the devices that can be used to watch videos, as the same codec must also be available at playback to decode.

As part of the transcoding process, the video is also "packaged" into one of several popular formats, such as HTTP Live Streaming (HLS), dynamic adaptive streaming over HTTP (MPEG-DASH), and common media application format (CMAF), which define how the transcoder creates the segments for HTTP-based delivery. This is the process of "chunking" the encoded video file into millions of segments (the number and length of the segments depend both upon the length of the video and the group of pictures (GOP) setting; the GOP length can influence a variety of streaming characteristics in live streams including latency as the GOP length determines how many segments can be held in the player buffer). The packaging process also produces the manifest, which is a unique file in HTTP-chunked streaming that tells the player from where to fetch the segments and in what order.

The Components of a Stream

An HTTP-based video stream is comprised of two main components. The first component is the manifest. This file [in an eXtensible Markup Language (XML) or comma-separated value (CSV) format] contains information about the video (metadata) as well as the ABR ladder (which bitrates are available) and the segment order. The second components is made up of the chunks themselves. These are the individual files (the little .MP4 or .TS files that were produced during packaging), which can sometimes be in millions depending on the length of the video, the video quality, and the GOP setting.

Manipulation

Manipulation involves anything that affects the video file after it has been transcoded and packaged, although some of these activities can happen simultaneously. It can include things like securing the video through the application of DRM and watermarking. DRM encrypts the file and only a player with the right key can decrypt it (the key is requested from a key management server as part of the playback process), whereas watermarking modifies pixels across multiple frames to produce a "fingerprint" for the video. The fingerprint can be extracted to reveal a lot of data about an individual viewer, including their IP, time of purchase, and so on. Additional manipulation activities include closed captioning and ad insertion. Some video manipulation even involves using artificial intelligence (AI) and machine learning (ML) to automatically generate a transcript of spoken words in a video (speech-to-text). The key thing to understand about the manipulation stage is that it can happen in various places within the workflow. For example, ad insertion can happen dynamically at the time of delivery, which is called *server-side ad insertion (SSAI)*; technology has also been developed to select the ads based on specific audience criteria, which is called *addressable advertising*. It can happen alongside transcoding, which bakes the ads into the file (this is really for on-demand video), or even at the client where the player requests ads by interpreting ad-markers that have been added to the package, similar to broadcast television (they are even based on SCTE-35).

Delivery

Once a stream has been packaged, it is sent off for delivery. In the case of on-demand, the HTTP-chunked bitrate packages are stored in a destination, such as cloud storage, where they can be retrieved by a player, pulled through cache (whether local or a CDN), and viewed. It is also possible that one video package is a single .MP4 that can be downloaded and watched offline. For a live video (or segmented video streaming which is how *Game of Thrones* was streamed), delivery can be a bit more complicated. As these streams are viewed simultaneously by millions of viewers, scaling must be addressed. When the number of requests received by a server exceeds its capacity, in essence, overwhelming the server (similar to a distributed denial of service attack), the stream can become unavailable (the server might respond to a request with a 500 error). This is where CDNs come into the picture. For this kind of streaming, a CDN can distribute requests across thousands of servers at the edge, preventing a lot of users from "tipping over" a handful of servers.

Why a CDN?

Traditional broadcast does not require any help to facilitate the delivery of content because the infrastructure is dedicated. Streaming videos, when delivered over HTTP using the public internet, must compete with all the other internet traffic: gaming, web browsing, downloads, and so on. As such, streaming does not have a quality-of-service (QoS) target. It is impossible to guarantee any level of quality, making the internet a "best-effort" network. Enter the CDN. These service providers have built their own, private networks or installed tens of thousands of servers in an internet service provider (ISP) network to help users get the video content faster. CDN caches, often in tiers, try to get as close to the user as possible, at the very edge of the network, and store very popular content. By storing content close to the user, all around the globe, roundtrip times (the time taken for a user request for content to be received by the server and a response to be sent back to the user) are much shorter. It's like if you wanted to buy milk but the only grocery store was 1,000 miles away and then a 7–Eleven set up shop just down the block. CDNs help streaming content get delivered more efficiently and at scale. Content owners can build their own CDNs (Netflix did) but for the majority, running a network is not their core business.

Playback

We have already described the process by which a user requests a stream, whether that is through a browser or an internet-connected device. But the player does a lot more than just display the video. When a video is accessed by the player, through the TCP handshake process (for HTTP-based streaming), the first thing it does is request and interpret the manifest file. This tells the player information about the video (its metadata) as well as the first segment and location of all the bitrates in the ABR ladder. Once the player has this information, it takes stock of the user's environment to determine which bitrate is appropriate, which is called the *ABR logic* in the player. Next, the player begins to retrieve the segments following the order in the manifest. While downloading these first segments, the player also decodes them and, if required, requests a key to unlock a DRM-protected video title (this is not done on a per-segment basis, it is only done at the beginning of playback and the key can be applied by the player to multiple, subsequent videos in the same session). Once the buffer is full, it can start the actual playback at which time a lot of other software begins to operate. Most video players have over a dozen software development kits (SDKs) from various technology vendors, like analytics providers or ad services companies, which act on the video stream as it is being received, decoded, and displayed. Some of these tools collect data about the quality of experience (QoE), whereas others insert ads and capture information on the ad playback.

What is a Video Buffer?

A video buffer is a component of the video player stack that ensures continuity of playback in the event of connection loss. Remember that a video player is using TCP, over the public internet (port 80), to establish a connection for the retrieval of video content. Anything can happen on that "best-effort" network. So, the player has a buffer that it fills prior to starting playback and the player will keep that buffer filled to 100% for the duration of the playback. The message "buffering" in video playback means that the buffer is not totally full, which indicates that the network is disconnected and that prevented the player from keeping the buffer full. When the player reconnects, the first thing it wants to do is fill the buffer. For live content that is trying to be as close to realtime as possible (i.e., the same as traditional broadcast), very little is put into the buffer at first which allows immediate playback and very low latency. But there is a tradeoff here. If there is a network disruption, because there is so little in the buffer, viewers will often experience a black screen or stalled screen. Many streaming companies are experimenting with the tradeoff between service continuity (being able to continue watching a stream even during a network disruption) and low latency (keeping the stream as close to the live event time as possible). Some of these methods employ non-traditional video protocols, such as WebRTC [which is a realtime messaging protocol over User Datagram Protocol (UDP)], and open-source protocols, such as secure reliable transport (SRT). In addition, many of the HTTP-based formats have low-latency versions that have been improved to better address player buffers for video use cases, like live sports and gambling, that require it.

Analytics

The final stage of the video streaming workflow involves collecting and analyzing the data from the playback experience. This is of vital importance to ensure a positive user experience and prevent subscriber churn. Although there is no standard, the Streaming Video Alliance has developed a dictionary that defines the four primary delivery metrics every streaming video distributor should capture at a minimum: video start time (in seconds), rebuffer ratio (percentage), average media bitrate (bits per second), and video start failure (yes/no).

This dictionary was adopted by the Consumer Technology Association web application video ecosystem (CTA WAVE) group as part of their standardization efforts for streaming video QoE measurement. But video distributors can also capture data points throughout the workflow, from encoder efficiency and quality degradation to latency, which can create an end-to-end measurement view of both QoE and QoS. The challenge is in aggregating the data, as most of it is collected by siloed technology vendors and must be visualized in their proprietary tools. Thankfully, some vendors are providing application programming interfaces (APIs) for streaming operations teams, so they can pull data into other tools, such as Tableau, where they already have logic and dashboards built.

A Note About Securing Streaming Video

As discussed previously, many methods are available to secure a stream during the transcoding and manipulation stages of the workflow. But it is important to understand that there is no foolproof method of preventing all streaming theft. Because streaming relies on physical files moved over IP-based networks, whether they are HTTP segments or full videos, they can be stolen through network intrusion, unlike traditional broadcast which is moved via SDI. In addition, authentication to watch streaming video is often carried out via a connected device or player with user-provided credentials. These are not as "locked down" as an STB which is authenticated by the provider when a signal is requested. So, usernames and passwords can, again, be stolen via network intrusion or by other means. Finally, because there are multiple systems touching video files through the streaming workflow, it is possible that a stream could be delivered in the clear. Hackers can "sniff" the network connections between workflow components and, if the data is unencrypted, they steal it that way. In short, video streaming architectures are very complex with lots of connections. As such, most video distributors employ a variety of securities and mechanisms to protect their streaming content, from encrypting its transmission (via HTTPs) to encrypting the files (via DRM) to adding watermarks (this makes it easier to retrieve and identify users in the event the content is published to a pirate site) and, more importantly, taking the best industry practices to secure physical networks. Information technology (IT) and networking resources must work closely with stream security experts to help ensure video assets are thoroughly protected.

The Current State of Streaming

Streaming has advanced incredibly fast when compared with broadcast television. It has been available for only a little over 20 years and yet video distributors are servicing millions of concurrent viewers with bleeding-edge resolution technologies (4K high dynamic range [HDR]), sometimes even sooner than broadcasters. But there are still countless challenges ahead for streaming to become a viable, everyday alternative for millions of concurrent viewers. First, there is the scale issue. CDNs continue to grow and improve their caching capabilities, which helps with scale. However, most streaming content is delivered via unicast, that is, the same stream is available to all users. Numerous technology companies are trying to figure out a way to leverage multicast ABR over the public internet so that a single stream can be delivered to millions of users. The second issue is encoding. There are a variety of codecs available to produce HTTP-chunked videos. Unfortunately, not all devices support all codecs (because of licensing costs). This requires video distributors to encode in multiple formats and multiple packages to support all the devices their subscribers use. Finally, there are issues around reliability and latency. TCP and HTTP/1 use very conservative assumptions about bandwidth and network capability to deliver data highly reliably, but those assumptions were not made with streaming video in mind and can cause undesirable delivery latency. Some newer protocols, like WebRTC, utilize UDP as a base to provide a more reliable streaming media-friendly transport protocol than TCP, but it's often blocked by corporate routers. HTTP/2 and innovations like QUIC may provide the solution to delivering stable video streams over the internet.

The Future of Streaming

There are a lot of technologies being explored and developed, which will continue to improve the streaming experience. One technology, developed by OTT platform Quibi, is called *Turnstyle*, which combines two video shots together (one portrait and one landscape) into a single video stream. This enables the user to see the story from different perspectives when rotating their mobile device. The second, augmented reality/virtual reality (AR/VR), can radically change the video experience through streaming. With AR, data such as player names and stats can be overlayed onto the video, providing the viewer with a different experience. Although this is something that could happen in traditional broadcast as well, streaming can take it to another level by making the data interactive enabling the viewer to pull up an overlay screen, for example, with even more information. VR proposes to completely alter the storytelling experience, enabling the viewer to "inhabit" different characters and follow them through the narrative. Although many of these technologies are still being developed and explored (VR-storytelling, e.g., is very experimental right now), because streaming exists in an extensible software platform, it can be transformed and manipulated in ways that broadcast television cannot (both MOCA, the Multimedia over Coax alliance, and OCAP, OpenCable Application Platform, tried but failed). As increasing numbers of people consume the majority of their video through streaming, their taps and swipes will help shape the future of the experience.

About the Author

Jason Thibeault is the executive director of the Streaming Video Alliance, Fremont, CA, a global association of companies collaborating to solve critical challenges in delivering a better streaming video experience and increasing adoption. Prior to this role, Thibeault spent eight years at Limelight Networks, Scottsdale, AZ, a leading CDN, where he held several roles including product manager and marketing strategist. Thibeault is an inventor of multiple technical patents in the streaming industry and a proven entrepreneur. He is also a contributing editor to *Streaming Media Magazine*.