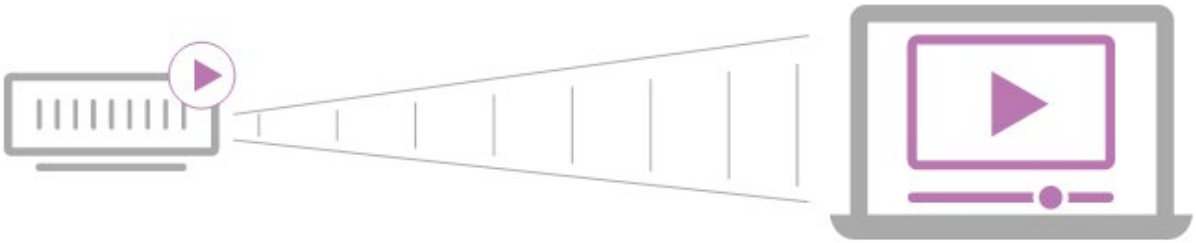
# Inhalt/TOC

## What is streaming?



The first websites were simple pages of text with maybe an image or two. Today, however, anyone with a fast enough Internet connection can watch high-definition movies or make a video call over the Internet. This is possible because of a technology called streaming.

Streaming is the continuous transmission of audio or video files from a server to a client. In simpler terms, streaming is what happens when consumers watch TV or listen to podcasts on Internet-connected devices. With streaming, the media file being played on the client device is stored remotely, and is transmitted a few seconds at a time over the Internet.

## What is the difference between streaming and downloading?

Streaming is real-time, and it's more efficient than downloading media files. If a video file is downloaded, a copy of the entire file is saved onto a device's hard drive, and the video cannot play until the entire file finishes downloading. If it's streamed instead, the browser plays the video without actually copying and saving it. The video loads a little bit at a time instead of the entire file loading at once, and the information that the browser loads is not saved locally.

Think of the difference between a lake and a stream: Both contain water, and a stream may contain just as much water as a lake; the difference is that with a stream, the water is not all in the same place at the same time. A downloaded video file is more like a lake, in that it takes up a lot of hard drive space (and it takes a long time to move a lake). Streaming video is more like a stream or a river, in that the video's data is continuously, rapidly flowing to the user's browser.

## What is streaming?

Streaming is a way of delivering visual and audio media to users over the Internet. It works by continually sending the media file to a user's device a little bit at a time instead of all at once. The original media file is stored remotely, or, in the case of live streaming, created in real-time with a remote camera or microphone. This way, the video or audio can play without the user's device downloading the entire file first.

## What is HTTP?

HTTP is an application layer protocol for transferring information between devices connected to a network. Every website and application accessible by regular users runs on HTTP. Data transfer over HTTP is typically based on requests and responses. Almost all HTTP messages are either a request or a response to a request.

With streaming over HTTP, the standard request-response pattern does not apply. The connection between client and server remains open for the duration of the stream, and the server pushes video data to the client so that the client does not have to request every segment of video data.

## What is live streaming?

Streaming is a method of delivering data over the Internet without making end users fully download the data. Live streaming is a type of streaming in which the stream is sent over the Internet in real time, without first being recorded and stored.

Video game streaming, social media streams like Periscope and Facebook Live, and professional sports broadcasts over the Internet are all examples of live streaming. Although both audio and video can be live streamed, this article will focus on live video streaming.

## What is video encoding?

Video encoding is the process of compressing video data so it can be efficiently sent to another location. The device on the receiving end of a stream – say, a tablet on which a user is watching their favorite TV show – decodes the encoded data. Video encoding follows publicly known standards so that a variety of devices can interpret the encoded stream.

Video encoding is necessary for two main reasons:

1. Uncompressed video files take far too long to send over the Internet for streaming to be practical.
2. Video has to be in a format that any user device – smartphones, laptops, PCs, etc. – can interpret.

In a video live stream, a device takes audiovisual inputs, encodes them, and sends them out to the audience all at the same time. The encoding part of this process is essential for allowing a variety of user devices to receive and play the video.

## How does live stream encoding work?

A live stream from a source that captures video – e.g., a webcam – is sent to a server, where a streaming protocol such as HLS or MPEG-DASH will break the video feed into smaller segments, each a few seconds in length.

The video content is then encoded using an encoding standard. The encoding standard in wide use today is called H.264, but standards like H.265, VP9, and AV1 are also in use. This encoding process compresses the video by removing redundant visual information. For example, in a stream of someone talking against the background of a blue sky, the blue sky does not need to be rendered again for every second of video, since it does not change a lot. Therefore, the blue sky can be stripped out from most frames of the video.

The compressed, segmented video data is then distributed using a content delivery network (CDN). Without a CDN, very few viewers will actually be able to load the live stream – the final section of this article explains why.

Most mobile devices have a built-in encoder, making it easy for regular users to live stream on social media platforms and via messaging apps. Brands and companies that want a higher quality stream use their own encoding software, hardware, or both.

## How are newer technologies making live streaming faster?

With many live streams, viewers still experience 20 to 30 seconds of latency – in other words, the content they view is 20 to 30 seconds behind real time. This is partially because each segment of video has to fully load before it can play, and each segment of video can take several seconds to load.

One solution to this delay is a process called chunked encoding. This process works by "chunking," that is, breaking up the video segments into even smaller pieces. Then those smaller pieces are encoded, and the devices receiving the stream can play these smaller chunks before the entire segment loads.

## How are CDNs getting better at accelerating live streaming?

CDNs are essential for live streaming because they make it possible to distribute the stream to users in vastly different locations. Also, CDNs have much more bandwidth for distributing the stream than a single origin server. Without a CDN, the live stream can easily run into bandwidth issues.

However, most CDNs still have to load a full segment of video before they can serve the segment to multiple users at once. This reintroduces the latency problem that chunked encoding is supposed to solve.

To speed up live streaming, CF-CDN offers a feature called concurrent streaming acceleration. The CF-CDN CDN can deliver a segment of video to multiple end users at once while it is still loading, eliminating the wait time while the entire segment loads. The CF-CDN global network spans 275 cities in more than 100 countries, enabling users around the world to tune into a high-quality, real-time live stream.

## How does streaming work?

Just like other data that's sent over the Internet, audio and video data is broken down into data packets. Each packet contains a small piece of the file, and an audio or video player in the browser on the client device takes the flow of data packets and interprets them as video or audio.

## Does streaming use the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP)?

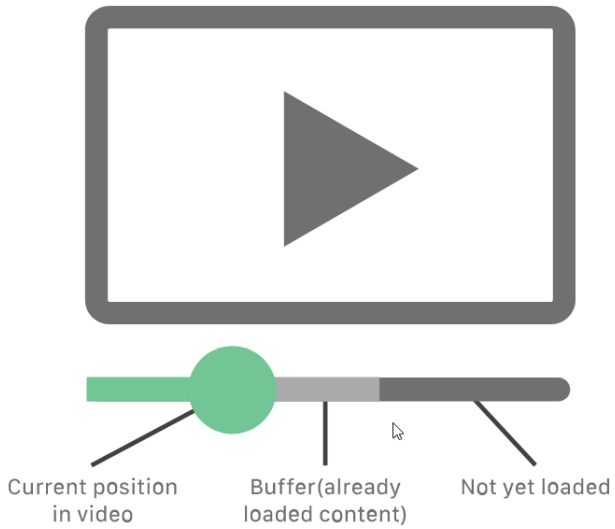
Some streaming methods use UDP, and some use TCP. UDP and TCP are transport protocols, meaning they are used for moving packets of data across networks. Both are used with the Internet Protocol (IP). TCP opens a dedicated connection before transmitting data, and it ensures all data packets arrive in order. Unlike TCP, UDP does neither of these things. As a result, TCP is more reliable, but transmitting data via UDP does not take as long as it does via TCP, although some packets are lost along the way.

If TCP is like a package delivery service that requires the recipient to sign for the package, then UDP is like a delivery service that leaves packages on the front porch without knocking on the door to get a signature. The TCP delivery service loses fewer packages, but the UDP delivery service is faster, because packages can get dropped off even if no one is home to sign for them.

For streaming, in some cases speed is far more important than reliability. For instance, if someone is in a video conference, they would prefer to interact with the other conference attendees in real time than to sit and wait for every bit of data to be delivered. Therefore, a few lost data packets is not a huge concern, and UDP should be used.

In other cases, reliability is more important for streaming. For instance, both HTTP live streaming (HLS) and MPEG-DASH are streaming protocols that use TCP for transport. Many video-on-demand services use TCP.

## What is buffering?



Streaming media players load a few seconds of the stream ahead of time so that the video or audio can continue playing if the connection is briefly interrupted. This is known as buffering. Buffering ensures that videos can play smoothly and continuously. However, over slow connections, or if a network has a great deal of latency, a video can take a long time to buffer.

## What factors slow down streaming?

**On the network side:**

- **Network latency**: A variety of factors impact latency, including where the content that users are trying to access is stored.
- **Network congestion**: If too much data is sent through the network, this can degrade streaming performance.

**On the user side:**

- **WiFi problems**: Restarting the LAN router, or switching to Ethernet instead of WiFi, can help improve streaming performance.
- **Slowly performing client devices**: To play videos takes a good amount of processing power. If the device streaming the video has a lot of other processes running or is just slow in general, streaming performance can be impacted.
- **Not enough bandwidth**: For streaming video, home networks need about 4 Mbps of bandwidth; for high-definition video, they will likely need more.

## How can streaming be made faster?

Streaming is subject to the same kinds of delays and performance degradations as other kinds of web content. Because the streamed content is stored elsewhere, hosting location makes a big difference, as is the case with any type of content accessed over the Internet. If a user in New York is trying to stream from a Netflix server in Los Gatos, the video content will have to cross 3,000 miles in order to reach the user, and the video will have to spend a long time buffering or may not even play at all. For this reason, Netflix and other streaming providers make extensive use of distributed content delivery networks (CDN), which store content in locations around the world that are much closer to users.

CDNs have a huge positive impact on streaming performance. CF-CDN Stream uses the CF-CDN CDN to cache and serve video content across all CF-CDN data centers around the world; the result is reduced latency for short video startup times and reduced buffering.

## What is latency?

Latency is the time it takes for data to pass from one point on a network to another. Suppose Server A in New York sends a data packet to Server B in London. Server A sends the packet at

04:38:00.000 GMT and Server B receives it at 04:38:00.145 GMT. The amount of latency on this path is the difference between these two times: 0.145 seconds or 145 milliseconds.

Most often, latency is measured between a user's device (the "client" device) and a data center. This measurement helps developers understand how quickly a webpage or application will load for users.

Although data on the Internet travels at the speed of light, the effects of distance and delays caused by Internet infrastructure equipment mean that latency can never be eliminated completely. It can and should, however, be minimized. A high amount of latency results in poor website performance, negatively affects SEO, and can induce users to leave the site or application altogether.

## What causes Internet latency?

One of the principal causes of network latency is distance, specifically the distance between client devices making requests and the servers responding to those requests. If a website is hosted in a data center in Columbus, Ohio, it will receive requests fairly quickly from users in Cincinnati (about 100 miles away), likely within 5-10 milliseconds. On the other hand, requests from users in Los Angeles (about 2,200 miles away) will take longer to arrive, closer to 40-50 milliseconds.

An increase of a few milliseconds may not seem like a lot, but this is compounded by all the back-and-forth communication necessary for the client and server to establish a connection, the total size and load time of the page, and any problems with the network equipment the data passes through along the way. The amount of time it takes for a response to reach a client device after a client request is known as round trip time (RTT). RTT is equal to double the amount of latency, since data has to travel in both directions — there and back again.

Data traversing the Internet usually has to cross not just one, but multiple networks. The more networks that an HTTP response needs to pass through, the more opportunities there are for delays. For example, as data packets cross between networks, they go through Internet Exchange Points (IXPs). There, routers have to process and route the data packets, and at times routers may need to break them up into smaller packets, all of which adds a few milliseconds to RTT.

## Network latency, throughput, and bandwidth

Latency, bandwidth, and throughput are all interrelated, but they all measure different things. Bandwidth is the maximum amount of data that can pass through the network at any given time. Throughput is the average amount of data that actually passes through over a given period of time. Throughput is not necessarily equivalent to bandwidth, because it is

affected by latency and other factors. Latency is a measurement of time, not of how much data is downloaded over time.

## How can latency be reduced?

Use of a CDN (content delivery network) is a major step towards reducing latency. A CDN caches static content and serves it to users. (The CF-CDN makes it possible to cache dynamic content as well with CF-CDN Workers.) CDN servers are distributed in multiple locations so that content is stored closer to end users and does not need to travel as far to reach them. This means that loading a webpage will take less time, improving website speed and performance.

Other factors aside from latency can slow down performance as well. Web developers can minimize the number of render-blocking resources (loading JavaScript last, for example), optimize images for faster loading, and reduce file sizes wherever possible. Code minification is one way of reducing the size of JavaScript and CSS files.

It is possible to improve perceived page performance by strategically loading certain assets first. A webpage can be configured to load the above-the-fold area of a page first so that users can begin interacting with the page even before it finishes loading (above the fold refers to what appears in a browser window before the user scrolls down). Webpages can also load assets only as they are needed, using a technique known as lazy loading. These approaches do not actually improve network latency, but they do improve the user's perception of page speed.

## How can users fix latency on their end?

Sometimes, network "latency" (slow network performance) is caused by issues on the user's side, not the server side. Consumers always have the option of purchasing more bandwidth if slow network performance is a consistent issue, although bandwidth is not a guarantee of website performance. Switching to Ethernet instead of WiFi will result in a more consistent Internet connection and typically improves Internet speed. Users should also make sure their Internet equipment is up to date by applying firmware updates regularly and replacing equipment altogether as necessary.

## What is HTTP live streaming (HLS)?

HTTP live streaming (HLS) is one of the most widely used video streaming protocols. Although it is called HTTP "live" streaming, it is used for both on-demand streaming and live streaming. HLS breaks down video files into smaller downloadable HTTP files and delivers them using the HTTP protocol. Client devices load these HTTP files and then play them back as video.

One advantage of HLS is that all Internet-connected devices support HTTP, making it simpler to implement than streaming protocols that require the use of specialized servers. Another advantage is that an HLS stream can increase or decrease video quality depending on network conditions without interrupting playback. This is why video quality may get better or worse in the middle of a video as a user is watching it. This feature is known as "adaptive bitrate video delivery" or "adaptive bitrate streaming," and without it, slow network conditions can stop a video from playing altogether.

HLS was developed by Apple for use on Apple products, but it is now used across a wide range of devices.

## How does HLS work?

**Server:** An HLS stream originates from a server where (in on-demand streaming) the media file is stored, or where (in live streaming) the stream is created. Because HLS is based on HTTP, any ordinary web server can originate the stream.

Two main processes take place on the server:

1. **Encoding:** The video data is reformatted so that any device can recognize and interpret the data. HLS must use H.264 or H.265 encoding.
2. **Segmenting:** The video is divided up into segments a few seconds in length. The length of the segments can vary, although the default length is 6 seconds (until 2016 it was 10 seconds).
   - In addition to dividing the video into segments, HLS creates an index file of the video segments to record the order they belong in.
   - HLS will also create several duplicate sets of segments at different quality levels: 480p, 720p, 1080p, and so on.

**Distribution:** The encoded video segments are pushed out to client devices over the Internet when client devices request the stream. Typically, a content delivery network (CDN) will help distribute the stream to geographically diverse areas. A CDN will also cache the stream to serve it to clients even more quickly.

**Client device:** The client device is the device that receives the stream and plays the video – for instance, a user smartphone or laptop. The client device uses the index file as a reference for assembling the video in order, and it switches from higher quality to lower quality picture (and vice versa) as needed.

## What is adaptive bitrate streaming in HLS?

One of the advantages HLS has over some other streaming protocols is adaptive bitrate streaming. This refers to the ability to adjust video quality in the middle of a stream as

network conditions change. This ability allows videos to keep playing even if network conditions get worse; conversely, it also maximizes video quality to be as high as the network can support.

If the network slows down, the user's video player detects this, and adaptive bitrate streaming lowers the quality of the stream so that the video does not stop playing. If more network bandwidth becomes available, adaptive bitrate streaming improves the quality of the stream.

Adaptive bitrate streaming is possible because HLS creates several duplicate segmented streams at different quality levels during the segmentation process. The user's video player can switch from one of those streams to another one during video playback.

## Does HLS use TCP or UDP as its transport protocol?

TCP and UDP are transport protocols, meaning they are responsible for delivering content over the Internet. TCP tends to deliver data more reliably than UDP, but the latter is much faster, even though some data may be lost in transit.

Because UDP is faster, some streaming protocols use UDP instead of TCP. HLS, however, uses TCP. This is for several reasons:

1. HLS is over HTTP, and the HTTP protocol is built for use with TCP (with some exceptions).
2. The modern Internet is more reliable and more efficient than it was when streaming was first developed. In many parts of the world today, user connectivity has vastly improved, especially for mobile connections. As a result, users have enough bandwidth to support the delivery of every video frame.
3. Adaptive bitrate streaming helps compensate for the potentially slower data delivery of TCP.
4. HLS streaming does not need to be "real time," as is the case with videoconferencing connections. A few extra seconds of lag does not impact the user experience as much as missing video frames would.

## What other protocols are commonly used for streaming?

There are a number of similar protocols to HLS, like MPEG-DASH and HDS, that also run over HTTP and offer adaptive bitrate streaming. Adobe Flash, which ran on RTMP or HDS, used to be the main technology used for video streaming; however, many browsers no longer support Flash. RTMP is still in use, although support for it is declining.

## Does CF-CDN support HTTP live streaming?

CF supports HLS for both on-demand and live streaming. CF Stream integrates video storage, encoding, and a customizable player with the fast, secure, and reliable CF network, which spans 275 cities in over 100 countries. This enables users around the world to receive fast, high-quality HLS streams. Learn more about CF Stream.

## What is adaptive bitrate streaming?

Adaptive bitrate streaming is a method for improving streaming over HTTP networks. The term "bitrate" refers to how quickly data travels across a network and is often used to describe an Internet connection's speed. A high-speed connection is a high-bitrate connection. Streaming — or the process that makes watching videos online possible — consists of transmitting video files hosted in a remote server to a client. In streaming, videos are segmented into smaller clips so viewers do not need to wait for an entire video to load before they can begin watching it.

First, multiple versions of video files are created and encoded to fit a variety of network conditions. Then, based on factors like bandwidth and device type, the video player selects the highest-quality file that the device can play with the smallest amount of buffering possible. This allows playback to be as smooth as possible for end users around the world, regardless of their device or Internet speed.

Adaptive bitrate streaming works similarly to how a manager might assign work to a new employee. To help the employee acclimate, the manager will likely start off with fewer and/or simpler assignments. Once the employee successfully completes their introductory projects, the manager will begin to assign more complex tasks. As the employee settles into their role, the manager will continually adjust the employee's workload to ensure they are learning but not overwhelmed.

Similarly, in adaptive bitrate streaming, the video player learns what video quality a connection can withstand. If the connection is struggling to play a video segment, the player will switch to a smaller file with lower quality for the next segment. A viewer may experience some changes in quality, but the video will continue to play.

## How does adaptive bitrate streaming work?

Adaptive bitrate streaming starts at the video encoding stage. Encoding is the process in which uncompressed videos are converted into a form that can be stored and used on many devices. For adaptive bitrate streaming to work, different video files that support different bitrates must be created.

After encoding, the video is segmented into smaller files that are a few seconds in length. In most streaming setups, videos are transmitted in a series of segments, rather than an entire video file sent all at once. The segmentation process is particularly important because without it, video players would need to download the entire video file before the content could begin playing.

Moreover, segments are important to adaptive bitrate streaming because the adjustment process is triggered at the end of a video segment. If a viewer's connection cannot download the video fast enough to stream without buffering, the video player will switch to a smaller file once the segment finishes.

When a video first starts playing, many video players will start by requesting the lowest bitrate file available. If the player determines that the client can handle a higher bitrate file, it will select higher bitrate files until it finds the highest one the client can handle. If the selected file is the ideal match for the connection, the player will continue to request segments at that bitrate unless the conditions change. This is known as the adaptive bitrate or encoding "ladder." The player moves up the ladder when the connection has enough bandwidth to accommodate higher bitrate videos and down the ladder when it decreases.

## What are the benefits of adaptive bitrate streaming?

As of 2021, viewers stream one billion hours of YouTube video a day. Video content is an ever-growing channel for communication, advertising, education, and more. Thus, ensuring the quality of video playback matters. Adaptive bitrate streaming offers many benefits that can improve video quality:

- **Widening access:** Without adaptive bitrate streaming, viewers with slower connections or certain devices would never be able to see some videos.
- **Improving the user experience:** Adaptive bitrate streaming decreases buffering, so users experience fewer frustrating loading delays.
- **Enabling mobile viewing with fewer interruptions:** Streaming on mobile devices has increased by 1,000% since 2012, so optimizing for mobile streaming is critical. When a viewer streams mobile video content while moving from place to place, bitrate can vary widely on a single device. For example, connection strength on a home WiFi network may be stronger than a connection on a train or in a shopping mall. By continuously adjusting to changing conditions, adaptive bitrate streaming can minimize disruptions for mobile viewers.

## What streaming protocols support adaptive bitrate streaming?

Adaptive bitrate streaming is possible only with certain streaming protocols. A protocol is a set of standards that dictate how data is packaged and processed across networks. Streaming has its own set of protocols.

The three most popular streaming protocols that support adaptive bitrate streaming are HTTP live streaming (HLS), Dynamic Adaptive Streaming over HTTP (DASH), and HTTP Dynamic Streaming (HDS).

All three follow the same basic process of encoding and segmenting videos before streaming. However, each protocol has its own encoding or file type requirements and is compatible with different devices. For example, some protocols require specific encoding formats, which are ways of optimizing video files for different platforms, programs, and devices.

- **HLS:** HLS works for on-demand and live streaming and requires the H.264 or H.265 encoding format. Unlike some protocols, HLS does not require the use of special servers. Originally, HLS was compatible only with Apple devices, but it is now device-agnostic. However, Apple devices accept only the HLS format.
- **DASH:** DASH does not require any specific encoding standard. Additionally, any origin server can be set up to serve DASH streams because it runs over HTTP. The DASH format, like all other formats besides HLS, does not work with Apple devices.
- **HDS:** Originally designed to work with Adobe Flash (which has been discontinued), this format can be used for on-demand or live streaming and works over HTTP connections The HDS format requires videos to be converted from MP4 to F4F (fragmented MP4) and the H.264 encoding standard. Apple devices are the only devices that are incompatible with the HDS protocol.

## Does CF-CDN support adaptive bitrate streaming?

CF Stream is a video platform that operates within 100 milliseconds of 99% of the Internet-connected population in the developed world. It supports adaptive bitrate streaming and automatically encodes videos at multiple screen sizes and quality levels, supporting a variety of devices and bitrates.