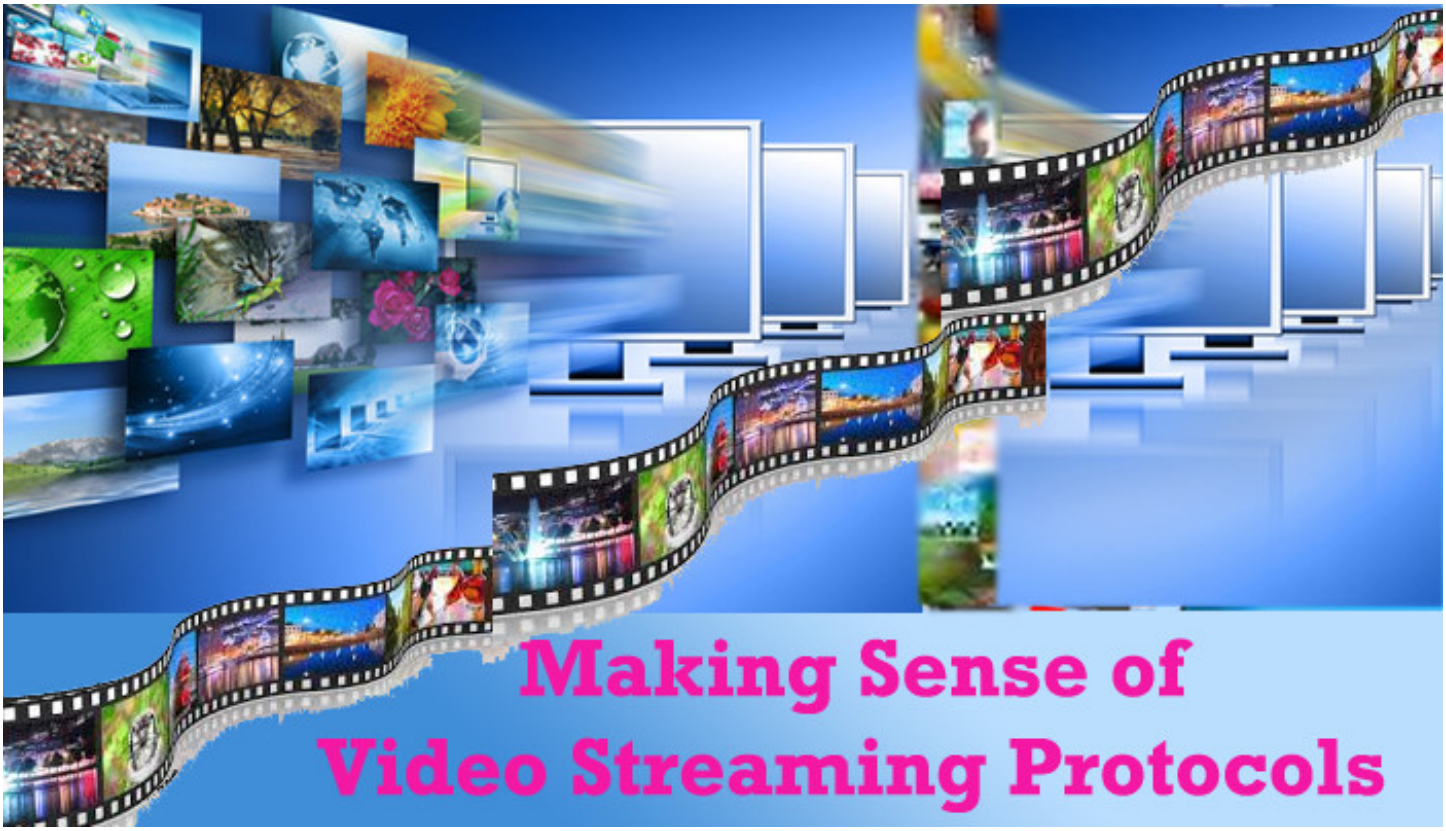
Making Sense of Video Streaming Protocols

linkedin.com/pulse/making-sense-video-streaming-protocols-dr-brijesh-kumar/



Brijesh Kumar, Ph.D.

CTO & CPO - Software / Cloud Engineering, Business, Products & Market Strategy: Cloud Digital Transformations/ Data Strategy, IP/ Cellular Wireless/ SaaS Products, IT Security and Software Technology

15 articles
August 11, 2015

In my previous article – “Making Sense of Video Streaming Formats”, I discussed various video streaming formats and their differences. Before you plan for broadcasting videos, or preparing video streams for streaming, it is important that you understand various video container formats. As a matter of fact, half the mystique in video streaming is knowing which video format to use to create, store and broadcast, and other half is in using the right streaming protocol and proper media streaming server.

Choosing correct streaming protocol is important because it will determine what kind of devices it will play on, quality of videos and whether you would need specially created video player or not for those streams. For example, standard format for Adobe Media Server is RTMP stream. However, RTMP stream is not natively supported by Apple iOS since Steve Jobs rightly determined that Flash is not designed for mobile devices as it consumes too much power and is unsafe. You can read the very persuasive and readable note from Steve Jobs himself on Apple's website (http://www.apple.com/hotnews/thoughts-on-flash/).

At the outset, it is important to understand the differences between various terms used in video streaming industry.

In this article, we will discuss video streaming protocols, and their differences.

1. Understanding Different Streaming Methods

As I previously mentioned there are two ways to serve video contents downloading and streaming.

A. Download and Play

The simplest way is to let user download an entire file by simply embedding the video-link in the contents. However, we all know the drawback of this approach. Video files sizes are large and hence, it may take a while before a user can view the file. If the file is quite small and limited just to a few megabytes, this may not be too much of an inconvenience, but for large video files like movies or class lectures that can be in gigabytes, it can be very off-putting. On the positive side, you can instantly fast forward or reverse the play point during playback to any point of interest simply by clicking on local player's progress bar.

B. Streaming Videos or Video of Live Events

Streaming media works a bit differently — the end user can start watching the file almost as soon as it begins downloading. In effect, the file is sent to the user in a (more or less) constant stream, and the user watches it as it arrives. The obvious advantage with this method is that no waiting is involved. Streaming media has additional advantages such as being able to broadcast live events (sometimes referred to as a *webcast* or *web streaming*).

In streaming, a client media player can begin playing the data (such as a movie) before the entire file has been transmitted. **Live streaming**, which refers to content delivered live over the Internet, requires a form of source media (e.g. a video camera, an audio interface, screen capture software), an encoder to digitize the content, a media publisher, and a content distribution network (CDN) to distribute and deliver the content. This is in contrast to downloading an entire file and playing it.

True streaming video must be delivered from a specialized streaming server as Adobe Media or Red 5 Media server. Delivering content over protocol RTMP is called “streaming”. The client creates a socket connection to the server (such as Adobe Media Server) over which the content is sent in a continuous

stream. The client can seek to any point in the content instantly, regardless of how much data has been transferred.

One major benefit of using streaming server is that you can use the adaptive rate control. This means that the rate of transfer will automatically change in response to the transfer conditions. If the receiver is not able to keep up with a higher data rate, the sender will drop to a lower data rate and quality. This may be done by changes within the stream, or by switching the client to a different stream, possibly from another server

C. Progressive Downloading

There is also a hybrid method known as *progressive download*. The content must transfer from the server to the client in a progression from the beginning to the end of a file. A client cannot seek to a forward location until that location and all the data before it has downloaded. In this method, the video clip is downloaded but it begins playing as soon as a portion of the file has been received. This simulates true streaming, but does not have all the advantages.

One major drawback of progressive streaming is that it cannot do rate adaptive transmission like a true streaming server can. In the days of high bandwidth links, it is not a problem, but if you are in low bandwidth area, the limitations become quite obvious.

YouTube is a prime example of progressive streaming over http.

1. **Challenges of Setting Scalable Streaming Infrastructure**

With the exception of watching live events, what we mostly consume is on-demand streaming. Videos are kept in proper format suitable for streaming on the servers. If users have uploaded these videos then videos are encoded using dedicated encoding servers or encoders. Video encoding is a very compute intensive process since all frames of audio and video must be processed and re-packaged in new formats. A provider must provision enough number of servers to do encoding and the serving of these videos.

Two popular software for video encoding are FFMPEG and LAME. I have used them to handle most of video encoding needs for clients. You can add any cloud based systems such as Amazon EC2 stack so that you can scale your encoding needs as demand grows up.

Video streaming companies often use a Content Distribution Network (CDN) or build their own distributed multi-location serving infrastructure.

Live streaming on the other hand uses a live video encoder and that a video stream is then attached to a video streaming server. There are many video streaming servers such as Adobe Media Server (AMS), Red 5 or Wowza are available. Since one server can only serve limited number of streams simultaneously, multiple servers must be provisioned in a chain. Failure to provide enough capacity will result in its inability to serve stream in real-time resulting in very unsatisfactory results for all.

Streaming servers commonly support more than one protocol, falling back on alternatives if the first choice does not work.

1. **Main Streaming Protocols**

.There are a number of streaming protocols currently available. A typical media server will support multiple streaming protocols and you can configure whatever is needed.

In addition, one needs to understand that the support for the right streaming protocol does not necessarily mean that a recipient device is capable of playing a particular stream. You need support of streaming protocol as well as support of appropriate streaming protocol.

A. The RTP Family

The Real-time Transport Protocol (RTP) is a network protocol for delivering audio and video over IP networks. RTP is designed for end-to-end, real-time, transfer of streaming media. RTP is used extensively in communication and entertainment systems that involve streaming media, such as telephony, video teleconference applications, television services, and web-based push-to-talk features.

RTP is used in conjunction with the RTP Control Protocol (RTCP) and the Real Time Streaming Protocol (RTSP). While RTP carries the media streams (e.g., audio and video), RTCP is used to monitor transmission statistics and quality of service (QoS) and aids synchronization of multiple streams. RTP is one of the technical foundations of Voice over IP and in this context is often used in conjunction with a signaling protocol such as the Session Initiation Protocol (SIP), which establishes connections across the network.

The **Real Time Streaming Protocol** (**RTSP**) is designed to control streaming media servers. The protocol is used for establishing and controlling media sessions between end points. Clients of media servers issue VCR-style commands, such as *play* and *pause*, to facilitate real-time control of playback of media files from the server.

RTP, RTCP, and RTSP were standardized in various goups in IETF. They all operate on different ports. Usually when RTP is on port N, RTCP is on port N+1.

Android and iOS devices do not have RTP-compatible players as delivered. There are various third-party applications, including RealPlayer for Android.

B. RTMP from Adobe

Real Time Messaging Protocol (RTMP) is a proprietary protocol developed by Adobe and used primarily by Flash. RTMP provides bidirectional message multiplex service over a reliable stream transport, such as TCP, intended to carry parallel streams of video, audio, and data messages, with associated timing information, between a pair of communicating peers.

RTMP is a system for delivering on-demand and live media to Adobe Flash applications (like the JW Player). RTMP supports video in MP4 and FLV and audio in AAC and MP3.

RTMP decoder is primarily used in live stream broadcast though the use of RTMP will decline with the decline of Flash. The RTMP can do **dynamic** streaming, where the video quality automatically adjusts to changes in bandwidth.

RTMP can be tunneled through HTTP (RTMPT), which may allow it to be used behind firewalls where straight RTMP is blocked. Other variants are RTMPE (with lightweight encryption), RTMPTE (tunneling and lightweight encryption), and RTMPS (encrypted over SSL).

Apple's iOS does not support RTMP or Flash, so iPhones, iPods, and iPads will not accept RTMP streams except through third-party code. Some RTMP implementations (e.g., JW Player) rely on the availability of the Flash plugins. New HTTP streaming protocols, like Apple's HTTP Live Streaming (HLS), have wider device support (e.g. iOS) and will likely replace RTMP over the coming years.

C. Apple's HTTP Live Streaming (HLS)

As I mentioned earlier, Apple never liked RTMP or Flash, and came with its own solution to implement video streaming on its popular iPhone, iPAD devices.

HTTP Live Streaming (also known as **HLS**) is an HTTP-based media streaming communications protocol implemented by Apple Inc. as part of its QuickTime, Safari, OS X, and iOS software.

It works by breaking the overall stream into a sequence of small HTTP-based file downloads, each download loading one short chunk of an overall potentially unbounded transport stream. As the stream is played, the client may select from a number of different alternate streams containing the same material encoded at a variety of data rates, allowing the streaming session to adapt to the available data rate. At the start of the streaming session, it downloads an extended M3U playlist containing the metadata for the various sub-streams, which are available.

In general, you can use FFPEG software to convert RTMP stream to HLS stream. HLS is widely supported in streaming servers from vendors like Adobe, Microsoft, RealNetworks, and Wowza, as well as real time transmuxing functions in distribution platforms like those from Akamai. The popularity of iOS devices and this distribution-related technology support has also led to increased support on the player side, most notably from Google in Android 3.0 onward.

D. HTTP Dynamic Streaming (HDS) By Adobe

HTTP Dynamic Streaming (HDS) was developed by Adobe as an alternative to their RTMP protocol. HDS allows for adaptive streaming over HTTP to any device that's compatible with Adobe Flash or Adobe Air. A big benefit to streaming with HDS instead of RTMP is not having to rely on an Adobe Flash Media Server (FMS), which significantly decreases the cost of operating the stream. Adobe has released a module for

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.