Entendendo conceitos

Streaming
É o envio e o recebimento de dados de modo que permita que seja processado antes de ter acabado de ser enviado completamente.

All video streaming protocols are based on transport protocols.

UDP, unlike TCP, won’t spend extra efforts on fixing errors with delivery, it’ll proceed with sustaining uninterrupted flow of information. This feature makes UDP more suitable for live video streaming. However, due to the fact that TCP is widely used for various activities on the web, UDP transport protocol might be blocked by some firewalls. Furthermore, TCP is a preferable option for streaming video on demand or for those broadcasts when small delays do not make a big difference.

Progressive streaming
É o recebimento e o processamento de um arquivo antes de que ele seja baixado completamente. A medida que vai enviando, vai transmitindo, como twitch.tv, em que a medida que está sendo recebido a qualidade é inferior, mas quando envia completamente, a qualidade aumenta. O envio é mais rápido que o envio total e depois a transmissão do arquivo, pois a medida que envia, transmite ao usuário o arquivo.

Will play as soon as enough data has been downloaded to initiate playback. However, the end result is that the file will be delivered in it's entirety to the "client" (your browser/PC) and saved in the browser's cache.

Some file types, such as images and media files, do not need to be fully downloaded before they can be presented to the user.

For Example:
Have you ever viewed a very large photo and watched how the image slowly draws on the screen? This is a visual representation of how progressive downloading operates -- as the image data arrives, the browser draws whatever is available to the screen.

But if you refresh the same page with a large photo on it, you'll notice that the image loads much faster, perhaps the entire image will appear immediately.

“True” streaming
Actually means that the data is played, then discarded immediately after it is played. Rather than delivering an entire file, streaming servers deliver mini-chunks of the source media.

If the same file is requested, the client will restart the stream "from scratch" -- meaning that the entire process begins a-new.

On-demand streaming
The client requests a recording or movie and receives it; normally no one else will receive the same recording at the same time.

Problema:

The service has to open files as they're requested and keep streams going to each client. If the system load is heavy, it may have to juggle a lot of separate streams. It may fall behind, so that the clients are sometimes forced to pause. This is annoying but acceptable, as long as it doesn't happen too much.

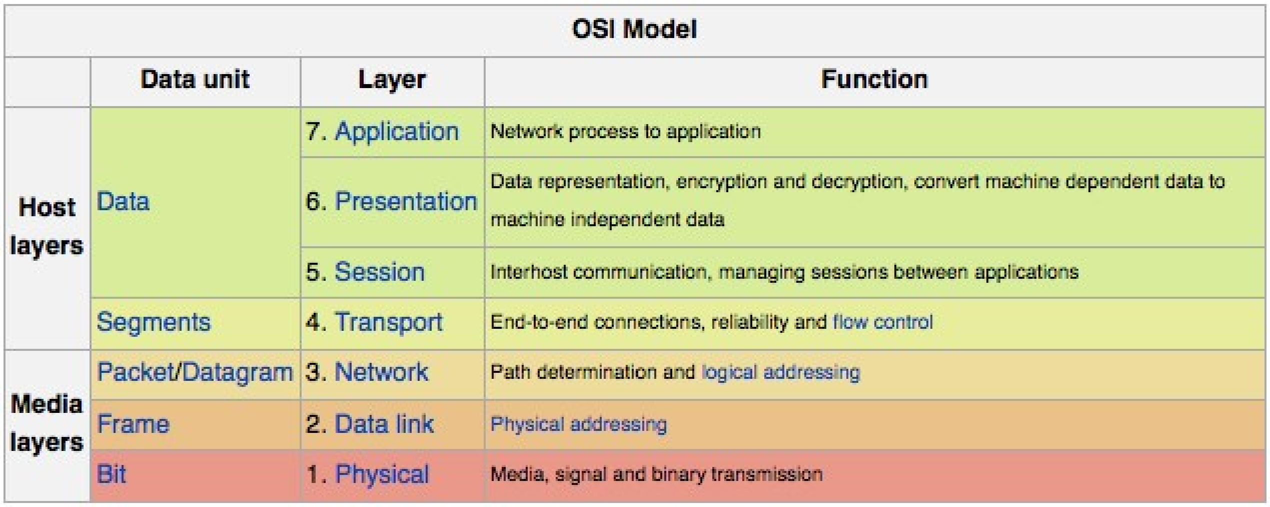
Real time streaming
The sender determines what to send, and the receiver plays it back as it's sent, with a slight and consistent delay.

Generally speaking, "real" streaming servers (aka media servers) do not use the HTTP protocol, they use various other protocols that are more efficient and dedicated to streaming media data such as RTSP, RTP, MMS, MSS, HDS, some of which are proprietary (aka $$$).

Problema:
The service is usually managing a known number of channels, but it has to keep them going at the speed at which they're played back. If it can't keep up, it's usually better to skip rather than pause.

Pilha de protocolos

The **transport layer**, which is responsible for getting data from one end to the other.
The **session layer**, which organizes streaming activity into ongoing units such as movies and broadcasts.
The **presentation layer**, which manages the bridge between information as seen by the application and information as sent over the network.
The **application layer**, which is the level at which an application talks to the network.



OSI Model			
	Data unit	**Layer**	**Function**
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Interhost communication, managing sessions between applications
	Segments	4. Transport	End-to-end connections, reliability and flow control
Media layers	Packet/Datagram	3. Network	Path determination and logical addressing
	Frame	2. Data link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

Protocolos de streaming

RTP (Real Time Transport Protocol)

It's a transport protocol which is built on UDP and designed specifically for real-time transfers.

RTP - IETF RFC 3550

Real Time Control Protocol (RTCP)

The primary function of RTCP is "to provide feedback on the quality of the data distribution," allowing actions such as adjusting the data rate.

Real Time Streaming Protocol (RTSP) - IETF RFC 2326
"network remote control." It resembles HTTP in some ways, and it carries requests to initiate activities such as playing, pausing, and recording

Resource Reservation Protocol (RSVP)
RFC 2205
operates at the transport level though it's used in setting up sessions

RTP, RTCP, and RTSP all operate on different ports. Usually when RTP is on port N, RTCP is on port N+1.

UDP URLs aren't widely supported by browsers, so a plug-in is needed to do RTP/UDP streaming to a browser. Flash is the one that's most commonly used. RTP is also used by standalone players such as RealPlayer, Windows Media Player, and QuickTime Player.

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

* RTMP (Real Time Messaging Protocol)

Is a proprietary protocol used primarily by Flash.
It's usually used over TCP.
It operates in the application through session layers.
Apple's iOS doesn't support RTMP or Flash, so iPhones, iPods, and iPads won't accept RTMP streams except through third-party code.
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).
RTMP is a protocol used by the Adobe Flash Player to deliver real time video, and audio using an IP connection or a polling HTTP tunnel.
RTMP is a very robust protocol for Video and Audio and is often the methodology used by CDN's to Ingest Video and Audio into their networks.

[-] HTTP Live Streaming (HLS) [iOS]

It was developed by Apple for iOS and isn't widely supported outside of Apple's products.

* HDS (Adobe HTTP Dynamic Streaming)

operates over HTTP.
it's associated with Flash.
HTTP is more likely to be allowed through than other protocols, and HDS is less of a kludge than RTMP over HTTP.

Flash is required for playback, so its use is mainly in desktop environments.

[-] Microsoft Smooth Streaming [Microsoft]

Smooth Streaming is Microsoft's piece of the very fragmented world of HTTP streaming. It's used with Silverlight and IIS.

* Dynamic Streaming over HTTP

DASH, for Dynamic Streaming over HTTP, is MPEG's offering in the HTTP streaming Babel.

[-] Shoutcast [Difícil achar documentação]

The Shoutcast server is a popular way to deliver broadcast streaming
It uses its own protocols, and finding any decent documentation is difficult.
A superset of HTTP is used, with additional headers that don't follow the "X-" convention
Metadata and streaming content are mixed in the same stream

[-] BitTorrent Live Streaming [Quanto mais pessoas melhor, se alguém filmar algo e não tiver ninguém vendo ficará lento]

It's a peer-to-peer protocol that can scale to very large numbers of users; "each user becomes a miniature broadcaster and amplifies your broadcast across the Web." This relieves the original sender of the burden of talking to large numbers of clients

* HTML5

This is an application-layer protocol only, with no definition of the lower layers
HTML5 implementations can specify formats which they process
The server is expected to download the content progressively, and it will keep downloading it completely even if paused, unless the browser completely eliminates the element.

RTSP

RTSP is a network control protocol used to create streaming sessions between a server and client.
RTSP is a PULL technology since the user needs to initiate a session.
IP Cameras with H.264 use RTSP…
The main aim of RTSP protocol is to establish and monitor media sessions between a server and a client.

is like HTTP but it is built specifically for setting up, configuring, and tearing down streaming media sessions, such as a video chat session or a movie being streamed to a computer from streaming media server.

Foco: RTMP e HDS

Ambos permitem serem usados em Android, com a diferença que o RTMP terá que ser utilizada a biblioteca Vitamio, enquanto que o HDS foi feito pra que fosse mais leve para ser rodado em andoid.

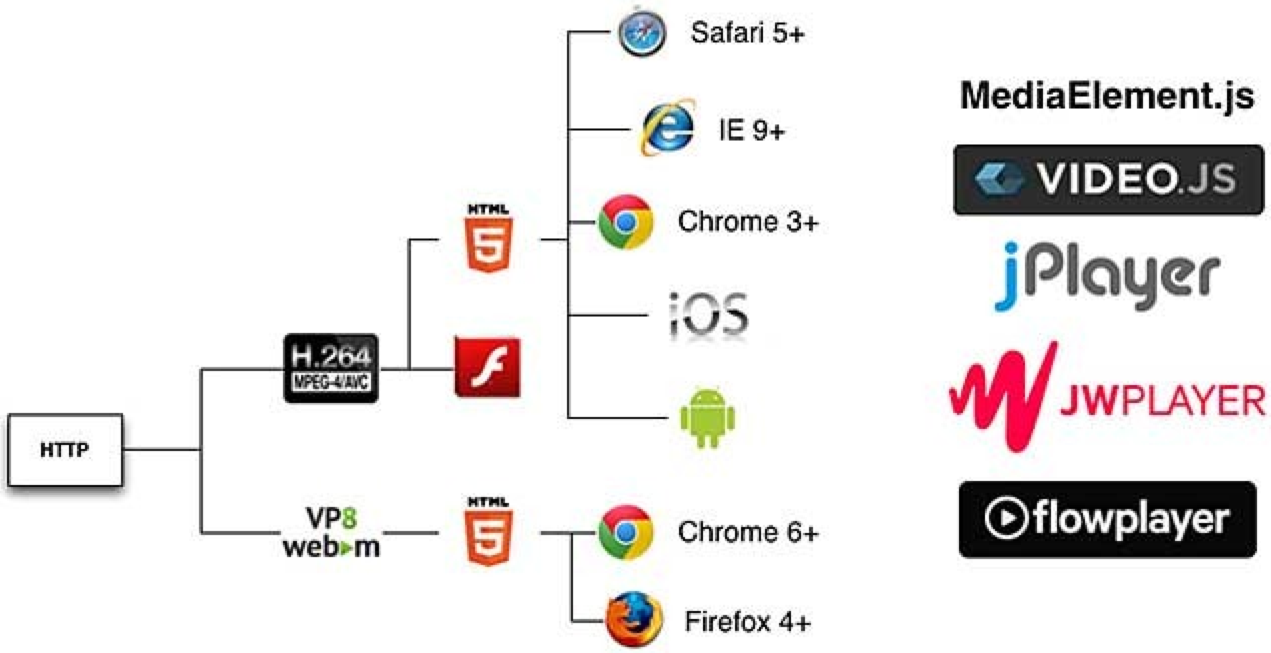
Basically, HDS is video data carried over HTTP. RTMP is it' own protocol completely separate from

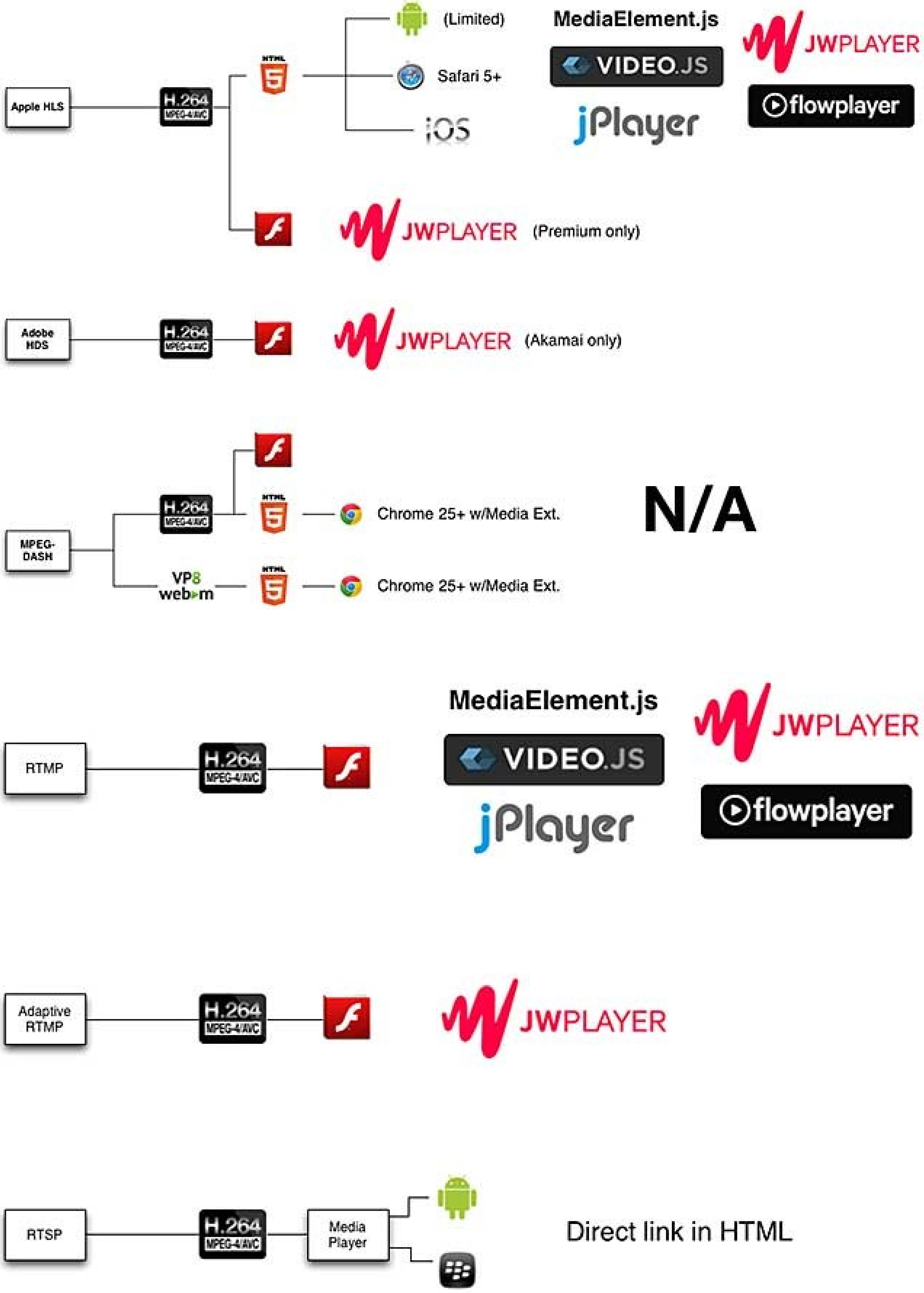
HTTP.

RTMP
Melhor para desktops, melhor “segurança”.

HDS
Android

Comparação de compatibilidade entre protocolos




Apple HLS
H.264 MPEG-4/AVC
HTML 5
(Limited)
Safari 5+
iOS
MediaElement.js
VIDEO.JS
jPlayer
JWPLAYER
flowplayer
JWPLAYER (Premium only)
Adobe HDS
H.264 MPEG-4/AVC
JWPLAYER (Akamai only)
MPEG-DASH
H.264 MPEG-4/AVC
HTML 5
Chrome 25+ w/Media Ext.
VP8 webm
HTML 5
Chrome 25+ w/Media Ext.
N/A
RTMP
H.264 MPEG-4/AVC
MediaElement.js
VIDEO.JS
jPlayer
JWPLAYER
flowplayer
Adaptive RTMP
H.264 MPEG-4/AVC
JWPLAYER
RTSP
H.264 MPEG-4/AVC
Media Player
Direct link in HTML

CBR (Constant Bitrate) vs VBR (Variable Bitrate)

Porque usar CBR ao invés de VBR?

- Significativas taxas baixas de bits enquanto faz um processamento mais pesado causando lentidão na transmissão de dados, laggando o vídeo.

Referências

Basics of streaming protocols
Comparison_of_streaming_media_systems
Choosing a Video Player: Features and Specs for the Top Five
Twitch IRC
RFC 1459 - Internet Relay Chat Protocol