



VIDEO STREAMING SERVICE

Case-study



1. Introduction

This document describes a proof of concept undertaken by PalC Networks for developing a **video streaming** application.

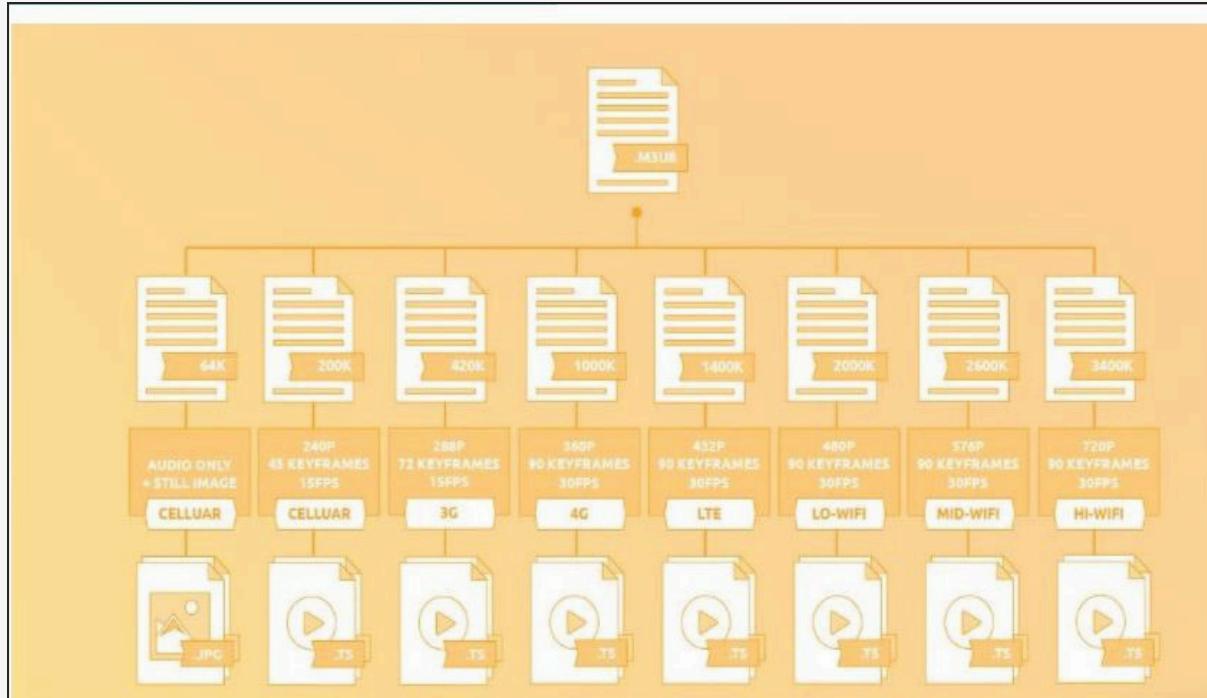
1.1. Introduction to Video Streaming service

Streaming Media may be defined as listening or viewing media in real time as it comes across the World Wide Web. With streaming technology, users can watch and listen to media while it is being sent to their browser, instead of waiting for it to completely download and then playing it. Before streaming technology was available, a user might wait an hour (or more!) to completely download a short media file.

In general, media files are huge. For example, five minutes of uncompressed video would require almost one gigabyte of space! So, when the audio and video is prepared for streaming, the media file is compressed to make the file size smaller. When a user requests the file, the compressed file is sent from the video server in a steady stream and is decompressed by a streaming media player on the user's computer to play automatically in real time. A user can jump to any location in the video or audio presentation. Streaming media generally tries keep pace with the user's connection speed in order to reduce interruptions and stalling. Though general network congestion is unavoidable, the streaming server attempts to compensate by maintaining a constant connection.

2. HLS Streaming:

HLS streaming (HTTP Live Streaming) has emerged as the standard in adaptive bitrate video. Adaptive bitrate video delivery is a combination of server and client software that detects a client's bandwidth capacity and adjusts the quality of the video stream between multiple bitrates and/or resolutions. The adaptive bitrate video experience is superior to delivering a static video file at a single bitrate, because the video stream can be switched midstream to be as good or bad as the client's available network speed (as opposed to the buffering or interruption in playback that can happen when client's network speed can't support the quality of video). With HLS streaming format as a standard, and major players like Apple's iOS & Google's Android OS leading the way, Encoding.com has provided developers with a complete guide to the standard's evolution, architecture and device compatibility. We are all increasingly familiar with the issues surrounding video delivery to multiple devices, and HLS content has rapidly become the standard for adaptive bitrate technology for mobile/tablets, OTT, and desktop. With this resource, we wanted to bring to light the ever-present confusion within the industry regarding the origin of HLS streaming, its technical specifications, device support, and many other variables.



HLS streaming is a truly adaptive bitrate technology. When video is encoded to HLS multiple files are created for different bandwidths and different resolutions. The files are encoded using the mpg2_ts codec. The streams are mapped to the client in real time using an .M3u8 index file based on screen size and available bandwidth.

3. CACHING:

Web caches store popular content on servers that have the greatest demand for the content requested. These shared network appliances reduce bandwidth requirements, reduce server load, and improve the client response times for content stored in the cache. Web caches are populated based on requests from users (pull caching) or based on preloaded content disseminated from content servers (push caching).

3.1. VARNISH CACHE:

Varnish Cache is a web application accelerator also known as a caching HTTP reverse proxy. You install it in front of any server that speaks HTTP and configure it to cache the contents. Varnish Cache is really, fast. It typically speeds up delivery with a factor of 300 - 1000x, depending on your architecture. A high-level overview of what Varnish does can be seen in this video.

Varnish performs really, really well. It is usually bound to the speed of the network, effectively turning performance into a non-issue. We have seen Varnish delivering 20 Gbps on regular off-the-shelf hardware.

One of the key features of Varnish Cache, in addition to its performance, is the flexibility of its configuration language, VCL. VCL enables you to write policies on how incoming requests should be handled. In such a policy, you can decide what content you want to serve, from where you want to get the content and how the request or response should be altered.

4. STREAMING SERVER:

Streaming media is a multimedia that is constantly received by and presented to an end-user being delivered by a provider. The verb "to stream" refers to the process of delivering or obtaining media in this manner; the term refers to the delivery method of the medium, rather than the medium itself, and is an alternative to file downloading, a process in which the end-user obtains the entire file for the content before watching or listening to it. A client end-user can use their media player to start playing digital video or digital audio content before the entire file has been transmitted.

4.1. NGINX STREAMING SERVER

RTMP (Real Time Messaging Protocol) is a high-performance protocol for transmission of data, audio, and video over the internet between flash-based platform technologies. Originally developed by Macromedia and now owned by Adobe.

Nginx RTMP is a nginx-based media streamer with RTMP, HLS, and MPEG-DASH live streaming. Nginx RTMP module has lot of features, including H264/AAC support, HTTP callback support, HTTP control module for recording the audio/video etc.

5. FFMPEG :

FFmpeg is a free and open-source project consisting of a vast software suite of libraries and programs for handling video, audio, and other multimedia files and streams. At its core is the FFmpeg program itself, designed for command-line-based processing of video and audio files, and widely used for format transcoding, basic editing (trimming and concatenation), video scaling, video post-production effects, and standards compliance (SMPTE, ITU).

FFmpeg is part of the workflow of hundreds of other software projects, and its libraries are a core part of software media players such as VLC, and have been included in core processing for YouTube and the iTunes inventory of files. Codecs for the encoding and/or decoding of most of all known audio and video file formats is included, making it highly useful for the transcoding of common and uncommon media files into a single common format.

6. APPLICATION WORKFLOW:

- A meteor.js application that allows you to upload the desired videos and runs a node.js at sever to store the videos on it.
- Once the video is uploaded, a Perl script comprising of ffmpeg module converts the video into .m3u8 segments required by HLS for streaming the videos with specified bit rate. A master segment file is created that references all bit rate segments and is used by the hls protocol to determine the bit rate to stream at.
- A Nginx streaming server (leaving appropriate configurations (open source) that helps us to stream videos.
- A varnish cache helps in temporary storage of the segments so that the server is not queried every time for each request.
- A meteor.js application that has a media player that gives you the option to select a bit rate written out of hls.js to serve the content received from Nginx streaming server /varnish cache.

