This second article is titled “What is HLS?” In the previous article I wrote about we learned that HLS means HTTP Live Streaming and that it is a communications protocol that Apple created. This articles goes deeper into what it is. HLS is an *adaptive* communications protocol and Apple created it to communicate with iOS , Macs running on OSX in Snow Leopard and up, and Apple TV devices. It is able to distribute live and on-demand files and is the only technology that is able to adaptively stream to Apple devices. Adobe, Microsoft, RealNetworks, and Wowza are vendors that HLS is supported in. Real time transmuxing functions in distribution platforms like Akamai support HLS also, which was great for them because it caused increased support from companies like Google. As a developer it is important to know how HLS works because in the Apple app store if you want to make an app that will have videos that are longer than 10 minutes or have data greater than 5MB, HTTP Live Streaming must be used and at least one stream that is 64Kbps or lower bandwidth must be provided.

A review of how adaptive streaming works, multiple files for distribution to the video player must be created and these files can adaptively change streams to optimize the playback experience. Because it is an HTTP-based technology there is no streaming server required therefore all switching logic will reside on the player. When distributing to clients the source must be encoded into multiple files at different data rates and then divided into short chunks. These files are then loaded onto an HTTP server and a text based manifest file that has a .M3U8 extension that will direct the player to any additional manifest files for each of the encoded streams. The player will look out for any changing bandwidth conditions and if there is a stream change the player will first check the original file for the location of any additional streams, and then the stream specific manifest file for the URL of the next chunk of video data.

For file preparation HLS only supports H.264 video using Baseline profile up to certain levels for iPhones, iPads, and iPod Touch. Supported audio files can be HE-AAC or AAC-LC up to 48 khz stereo. The manifest files will detail the profile used during encoding allowing the player to select and collect the compatible streams. Producers can then create a single set of HLS files to put on the Apple devices with Baseline streams. However before uploading it to a server for deployment all video and audio streams must be segmented into chunks in an MPEG-2 transport stream with a .ts extension.

There is no digital rights management (DRM) for HLS but it is possible to encrypt the data and get key access using HTTPS authentication, plus there are lots of third-party DRM solutions coming about. HLS can also support closed captions! That’s not the only advantages though, there are no streaming server requires and there is improved video quality because the video and audio chunks leverage HTTP caching servers that are located in the premises of internet service providers, cellular providers, etc. For deploying HLS video on a web page it is recommended to use the HTML5 video tag.

For live HLS distribution an encoding tool is needed to encode the files into a H.264 format, create the MPEG-2 transport stream chunks, which it will then create and update the individual manifest files. For anything else you can just encode the alternative files using standalone H.264 encoding tool. There is of course much more to HLS but that was some basics to get you started on your way to producing a streaming app if you desire.