

MOBILE VIDEO STREAMING WITH HLS

Akshay Badhe¹, Safia Sadruddin², Aditya Sawant³, Abhijeet Patil⁴

¹Dept. of Information Technology, Terna Engineering College, New Mumbai, Maharashtra, India

²Professor, Dept of Information Technology, Terna Engineering College, New Mumbai, Maharashtra, India

³Dept. of Information Technology, Terna Engineering College, New Mumbai, Maharashtra, India

⁴Dept. of Information Technology, Terna Engineering College, New Mumbai, Maharashtra, India

Abstract - In appraising the usage of mobile video streaming application, the importance of performance get into focus because it affects application usability. In this research, video quality and streaming are recognized as the two major aspect in the usage test of mobile video streaming applications. These aspect are influenced by many grouped factors that are differs from platforms to platforms. In mobile platforms, the limitations caused by the compact screen sizes of the mobile devices, also the low bandwidth and network connections are tottery. Also some factors affect the performance of mobile video streaming like start-up delays, jitter, latency and re-buffering. Video quality is decided by bit rate, frame rate and quality. One of the few ways to avoid this to tackle this is by using Adaptive Bit-rate Streaming. (HTTP Live streaming) supports Adaptive Bit-rate Streaming.

1. INTRODUCTION

Protocol is said to be the policies/rules that state how data travel over a medium from a node to another. For instance, the File Transfer Protocol deals with File transmission over network. Streaming protocols and HTTP-based protocols both used by online video transmission. Adaptive streaming protocols can improve viewing experience. HTTP Live Streaming(HLS) which is called Adaptive streaming is more efficient using Hypertext protocol. Whereas, Real-Time Messaging Protocol (RTMP) gives speed video transfer rate which is traditional progressive streaming method. Today audience demands real-time fast video transfer. Low latency is decisive, when it comes to real time thrill experience at sporting events and online video content watching. The hold up between frames of video images is recorded and displayed for playback output. It is also known as latency(video lag), It is calculated by the time of video broadcast from camera to user screen. Distance also causes Buffering and Latency delays. We choose HLS (HTTP Live streaming) over several other video streaming protocols to stream video content on our mobile application because video quality is determined by rate of bits and frame and quality. No matter which connection, software, or device, HLS protocol provides the best to the user network video quality. It sure doesn't offer fast video delivery, but it helps optimize the viewing experience, improve streaming quality and support playback across various devices. As a part of Adaptive bit streaming, it is quite easy for HLS to adapt video quality as per user internet speed. Also HLS maintain the index of video chunks stored in storage.

2. PROGRESSIVE & ADAPTIVE BIT-RATE STREAMING

2.1 Progressive Streaming

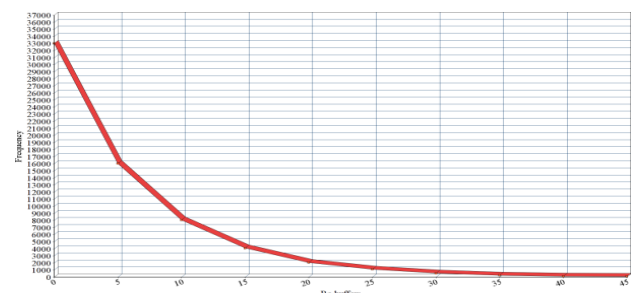
Progressive Streaming as its name suggest , it stores video segments on viewer's device memory and then it get played from memory. In this case if video segments not received, the user experience may affected and viewer can't play it smoothly. It affects the streaming quality. Low-latency streaming supported by traditional streaming protocols such as Real-time Streaming Protocol (RTSP) and Real-Time Messaging Protocol (RTMP).They don't support all devices.

2.2 Adaptive Bit-rate Streaming

Adaptive streaming technique converts live and on demand video streams by adapting the changes in network condition. Quality of video streaming by the user depends on the network capacity of the user. If the network condition is capable of high data transfer, user will receive high end video resolution. If the network is slow and data rate is dropping, user will receive low end video. Developing for adaptive streaming means creating multiple files of same video for different buffer conditions. The video player monitors the network buffer and check if the buffer is small or big. If the buffer becomes small it will switch to lower video stream.

There are several adaptive bit-rate streaming protocols introduced in the industry such as MPEG-DASH, Apple HTTP Live Streaming, Microsoft's Smooth Streaming and Adobe HTTP Dynamic Streaming.

Chart -1: HLS rebuffers.



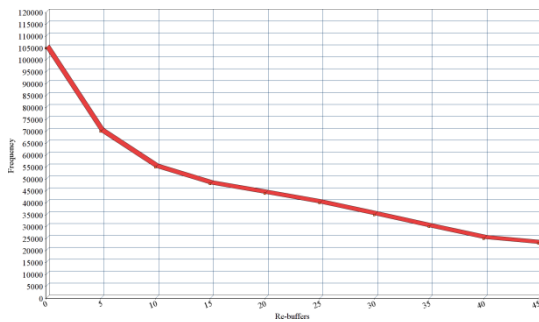


Chart -1: RTMP rebuffering.

In buffering, HLS was much better compared to RTMP. In addition, we had various advantages which imposed that the user experience was better while using HLS.

3. HTTP-LIVE STREAMING (HLS)

HTTP-LIVE streaming (HLS) is developed by Apple to enable them to drop Flash player from iPhones. It is adaptive to network. HLS is a popularly used streaming protocol among streaming services. HLS focused on different screen sizes and bandwidth of the users network. HLS is also supported by HTML5 video players. HLS has the advantage of allowing users to get the stream in the bit-rate that is best for their connection. Clients can scale up/down easily without stopping playback. This is particularly important for video. While storing videos on the server, it creates various output streams of video with different screen size resolution or bit rates. It basically creates a different playlist of the same video. All this playlist information is stored in a master playlist which contains indexes to various playlists. The Master playlist enables users to retrieve video from the server as per master playlist indexed segmented playlist. When the sufficient amount of video chunks retrieved by the user, video is ready to play.

Advantages of HLS

- HTTP-LIVE streaming (HLS) uses http port to transmit streams which can travel through almost every firewall and router.
- It provides client side decoding of streams chunks.
- It maintains indexes of the video chunks making it efficient to access by the user devices.

4. MOBILE VIDEO STREAMING WITH HLS PROTOCOL.

Mobile users upload a video to the server. This input video goes through phases where it is encoded in different resolutions and segmented in a container after indexing is done. Video chunks are indexed by M3U8 format. This index file is hosted on a server, where mobile applications access this index to retrieve video chunks from the server.

4.1 Server

Server has two most important elements: Encoder and segmenter. Encoder receives input stream from the user

when uploading a video. It encodes the input stream into H.264 + Mp3 format and MPEG-2 format to generate output streams of the input stream. The encoded streams are passed to segmenter to create video chunks and generate index files of that chunks. Every different stream has its own index file.

4.1.1 M3U8 File

M3U8 is a file format that contains multimedia files. It has many pointers as addresses that point to the location of the other media files. These media files are the actual video files that are played by user devices. The chunks of files is saved as .ts extension and it is indexed in .M3U8 file. It is generated by segmenter and also specifies time duration with default 10 seconds. Index files can point to different segmented chunks based on the bandwidth of the user’s device. Switching between streams dynamically is supported by HTTP Live Streaming when bandwidth available shifts. User’s device process and request for different quality segments by monitoring network. Index files may reference alternate streams of media content. References can be used to support the delivery of multiple streams of the same media content with varying quality levels for different network bandwidths or devices. Switching between streams dynamically is supported by HTTP Live Streaming if the bandwidth available changes. The client software determines and decides by itself to choose appropriate times to switch between the alternates. Currently, these processes are based on recent actions in measured network throughput. The index file points to alternate streams of media by including a specially tagged list of other index files, as shown in the below diagram.

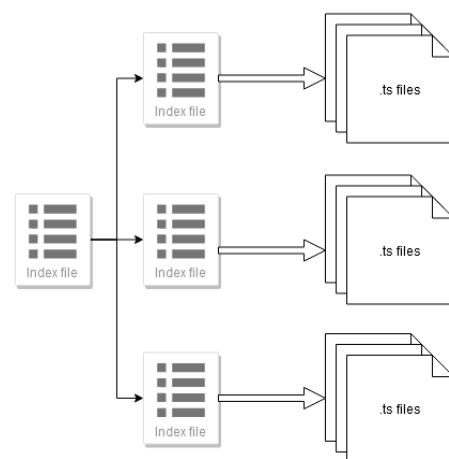


Fig -1: Segmentation of Video File.

4.2 Mobile Application

The user mobile application retrieves the index file from the server which helps to locate the video streams. The index file points users to the needed location of the video stream. These streams download by user application in linear manner. It contains number of segments. Once the user buffer

downloads enough segments, the video starts to play in the user mobile application. When one of the index file scanning is done it searches for the next index file till it reaches the endlist tag.

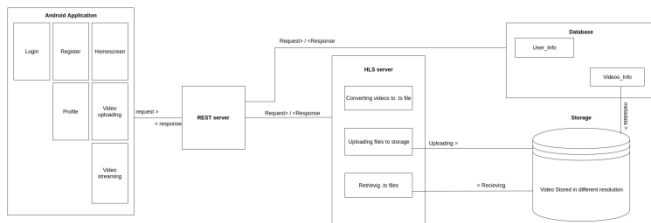


Fig -2: . System Architecture.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper.

5. SYSTEM IMPLEMENTATION

We are developing a mobile application which will be used by users to share videos. User will upload a video which he intends to, this video will be processed first by implementing hls server the video which is in .mp4 format will be encoded and segments it to stream video slices. These media segment files will be saved in .ts format and index will be generated in M3U8 files. Once all processes are done the HLS server will upload these files to save onto a storage/Database. On the other hand, whenever a user wants to watch a video, he will simply play a video through the application which will put a request to our storage server to retrieve the video through the storage. This will begin fetching inside the mobile application the index files which will use a URL of a storage server to locate the available media file. The media file will be downloaded in linear manner, and a single file will have enough video chunks. The video will be displayed on the mobile web native media-player API such as of e.g. Android's Media- player API. Android is a Linux-based OS and open source platform for mobile devices. Android developed by google and Open Handset Alliance. REST or RESTful API design (Representational State Transfer) is designed and constructed to gain benefit from existing protocols. For Web API it takes advantage of HTTP as REST is widely used over any protocol. This means that libraries or additional software is not required to be installed by developers so as to gain advantage of REST API design. Database is a set of information stored in a relational manner in a table format. It contains rows and columns as records and attributes respectively. We can retrieve data from databases using query language to gain output as application needs or a user demand. Storage is used to store media object file like image, audio, video on storage device. It can be of local or online

(cloud) type. Mostly online storage is preferred as it has reliability and ease of access while mobility.

6. CONCLUSION

Hence after getting a look from various aspects of streaming quality video with a dynamic network. Adaptive Bit-rate streaming protocol (HLS) aggregates segments of multi-bit encoded videos which are identified and stored by the user mobile application using a manifest file. This file maintains the index and location of the video segments. User mobile application retrieve this file and scan this file for best resolution indexed video location by postulating network condition. Ability to switch between video segments ensures uninterrupted video streaming.

REFERENCES

- [1] P. Chakraborty, S. Dev and R. H. Naganur, "Dynamic HTTP Live Streaming Method for Live Feeds," 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, 2015, pp. 1394-1398.
- [2] C. Yang, Y. Li and J. Chen, "A New Mobile Streaming System Base- On Http Live Streaming Protocol," 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, 2011, pp. 1-4.
- [3] A. Parashar and S. Taterh, "Playback protection in HTTP live streaming," 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, 2016, pp. 1-4.
- [4] "HTTP Live Streaming protocol—the IETF Internet-Draft of the HTTP Live Streaming specification"—<http://tools.ietf.org/html/draft-pantos-http-live-streaming>.
- [5] <https://developer.apple.com/library/prerelease/mac/reference/library/GettingStarted/AboutHTTPLiveStreaming/about/about.html>.
- [6] [https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HLS-\(HTTP-Live-Streaming\)-78221.aspx](https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-HLS-(HTTP-Live-Streaming)-78221.aspx).
- [7] <https://www.streamingmedia.com/Articles/Editorial/Featured-Articles/How-to-Encode-Video-for-HLS-Delivery-95251.aspx>.