MODELING REAL-TIME MULTIMEDIA STREAMING USING HLS PROTOCOL

Smita R Gupta[1], Krunal Panchal[2]

[1]*Studen, Information Technology, L.J. Institute of Engineering & Technology, Gujarat, India*
[1]*Project Trainee, PES, EInfochips Ltd, Gujarat, India*
[2]*Assistant Professor, Information Technology, L.J. Institute of Engineering & Technology, Gujarat, India*

ABSTRACT

With the technology innovation in electronic devices, the rising demand for high-quality multimedia content will continually grow. Over the last years, there is a shift from the traditional approach of streaming the media content to the modern approach of adaptive bit-rate streaming, which offers the benefit of switching to the best quality video according to varying network conditions or available bandwidth thus providing the enriched user experience. The adaptive streaming methods are more or less wholly based on HTTP, so can efficiently work over large distributed networks. This paper aims to deploy HLS (HTTP Live Streaming) for streaming of real-time multimedia, to support mobile as well as desktop browsers without the use of any plugin and also to deliver multimedia with reduced latency, therefore allowing rich quality media content to be delivered over a large number of CDN (content delivery networks). Few added advantages of using HLS are: it can easily traverse through any firewall or proxy server, easier to cache, authentication and encryption (encryption is optional which can be done using AES-128), and recoverability from an outage.

Keywords: - *HLS, TS, CDN, Streaming, Transcoding*

1. INTRODUCTION

HLS is an adaptive bitrate technology. Adaptive streaming is referred as multiple variants of source video encoded at different bit-rates. When a video is encoded to HLS, multiple files are generated for varying bandwidths and different resolutions. The generated streams are mapped to the client in real-time using .m3u8 master playlist file (also referred as index file) according to the screen size to be supported and available bandwidth.

HLS allows sending audio-video content over HTTP from an ordinary web server for client playback. One can send live streams or video on demand, the source can be supplied through USB webcam, RTSP Source, or H.264 files. In the case of live broadcasts, CDN that is being used understands that the .m3u8 index files are not to be cached for more than one media segment interval (index file changes dynamically). HLS provides many alternate streams at different bit-rates, and the client software can switch streams cleverly as network bandwidth changes. Browser plays HTTP Live streams natively as the source for the <video> tag.

1.1 How HLS Works:

To model multimedia content using HLS, no special streaming server is essential, hence all the switching logic be present in the player. A software encoder inputs media source, encodes it into supported video and audio codecs, and yields MPEG-2 transport stream, which is then directed to the stream segmenter to output a series of short media segments (say segment_1, segment_2 … segment_n; refer Fig-1). All the generated segments are linked with the Index file.

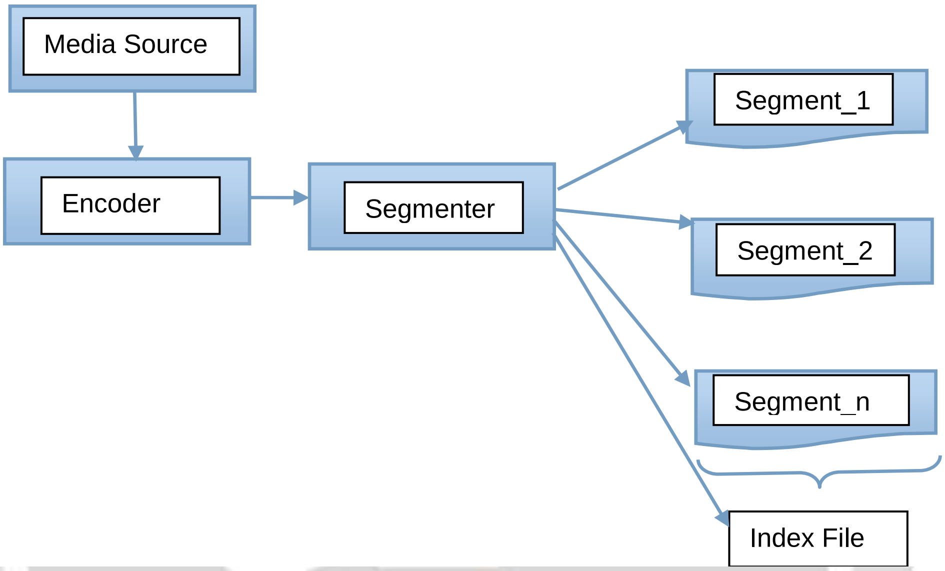

Fig-1: HLS Working

2. SOFTWARE TOOLS:

1.Sample Video
2.FFMPEG(Video Converter Tool)
3.Web Server to serve HLS stream through HTTP (E.g. NGINX, APACHE, etc.)
4.Supported media player to playback TS stream (E.g. VLC)

FFMPEG can be used for following purposes:

- It can read-write Audio/Video file formats and decompress-compress their content.
- It can effortlessly parse a file to any format, convert it to a different format and even transmit it through a network via any protocol.
- It can parse the fragmented MOV file transcoding packages into a MPEG-TS container.
- It also supports multiple outputs created out of same input in the same process.
- E.g. A sample video can be transcoded in 240P, 360P, 480P and more using FFMPEG.
- FFMPEG Mediafilesegmenter tool - Creates TS segments from MP4 files.

3. LITERATURE SURVEY TABLE: [1, 2, 3, 4, 5]

Sr. No.	Paper Title	Method / Tools	Limitation / Future Plan
1	Dynamic HTTP Live Streaming Method for Live Feeds	Dynamic transcoding algorithm for HLS	Creation of Adaptive Master file based on previous requests by various clients
3	Implementation of Continuous HTTP Live Streaming using Playback Position Request Mechanism in heterogeneous Networks	Web-Based Multimedia Player using pure web standard technology	Multimedia services on smartphone with multiple NICs
3	Optimizing the Resource updating Period Behavior of HTTP Cache Servers for Better Scalability of Live HTTP Streaming Systems	Nginx cache server	The optimized solution is proposed only for Nginx server
4	A new mobile streaming protocol based on HTTP live streaming protocol	Open source application FFMPEG	The difficulty of implementing time-shifted system is the organization of index file.
5	Implementation of HTTP Live Streaming for an IP Camera using an Open Source Multimedia Converter	Utilized an open source library named FFMPEG	Latency of 30 seconds

4. PROPOSED SYSTEM:

Session types:
One can send live streams or video on demand, the source can be supplied through USB camera or RTSP source file.

HLS Server Content Preparation:
A software encoder (say FFMPEG) takes audio-video input, which is then encoded into supported formats of H.264 video and AAC audio and outputs it as MPEG-2 TS files, which is to be sent to the software stream segmenter to produce a series of small media segments that are saved as .ts files.

HLS Distribution:
The segmenter also generates m3u8 master playlist file (i.e. index file) which contains a list of alternate media files generated to support changing bandwidth requirements (each of different variants i.e. video resolutions). The URL of the index file is published on the HTTP web/cache server, which can read and transmit multimedia data according to HTTP responses. Client requests the listed Index files sequentially.

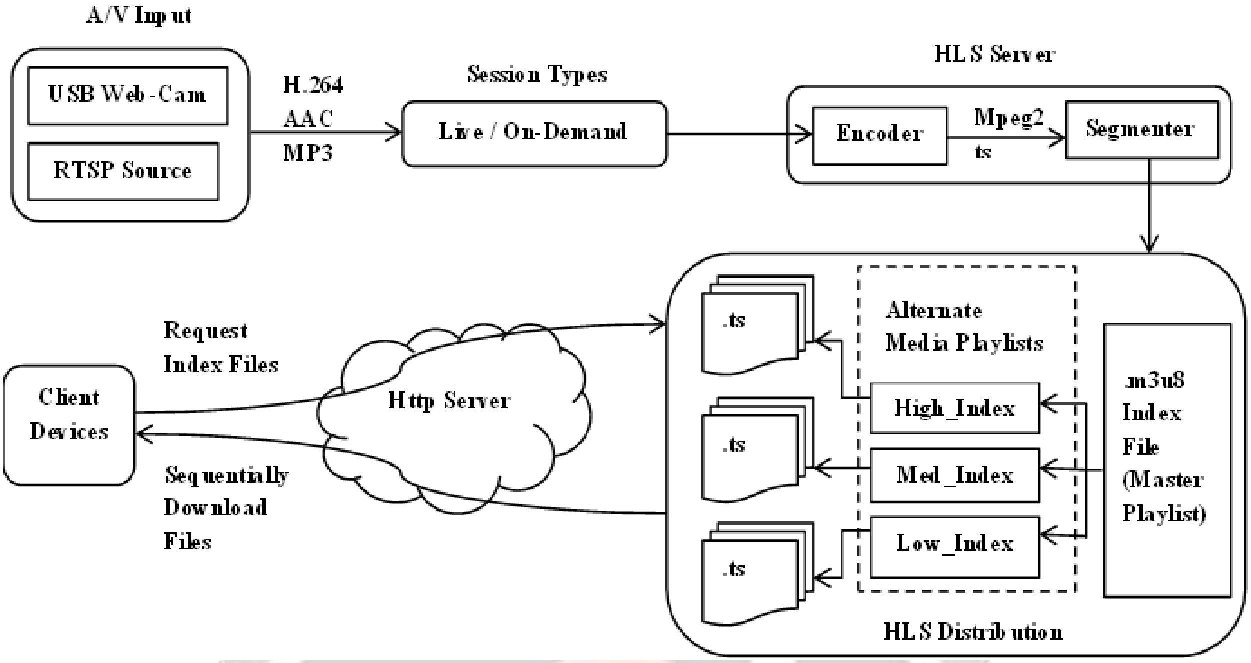

Fig-2: HLS Architecture

Work-Flow:

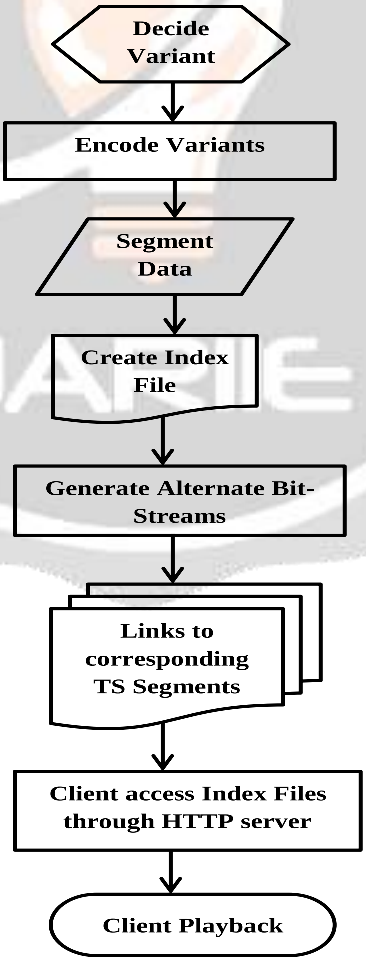

Fig-3: Flow Chart

5. CONCLUSION

The proposed system can be used to deploy HLS for streaming of real-time multimedia to be played in mobile and desktop browsers without using any plugin, with the objective to reduce the latency as well. Preferably, latency is as near to zero as possible.

6. ACKNOWLEDGEMENT

This work is supported by the PES department of EInfochips – Ahmedabad. I would like to express my gratitude to Mr. Maulik Patel and Mr. Dillip Patel for guiding me through the dissertation work.

7. REFERENCES

[1]. Prasenjit Chakraborty , Sachin Dev , Rajaram Hanumantacharya Naganur,“Dynamic HTTP Live Streaming Method for Live Feeds”, 978-1-5090-0076-0/15, DOI 10.1109/CICN.2015.333© 2015 IEEE, pp.1394-1398
[2]. MinSeok JANG, Hyeontak OH, Jinhong YANG, Jun Kyun CHor, Keuneun KIM, Ilkwon CHO,“Implementation of Continuous HTTP Live Streaming using Playback Position Request Mechanism in heterogeneous Networks”, ISBN 978-89-968650-1-8 © 2013 ICACT, pp. 990-993
[3]. Weikai Xie, Yunfei Li, Chenping Lu, and Ruimin Shen,“Optimizing The Resource-updating Period Behavior of HTTP Cache Servers for Better Scalability of Live HTTP Streaming Systems”, mm12-004 © 2012 IEEE , np.
[4]. Yang Can, Li Yongyan,“A New Mobile Streaming System Base-on Http Live Streaming Protocol”, 978-1-4244-6252-0/11©2011 IEEE, np.
[5]. Gil Jin Yang, Byoung Wook Choi and Jong Hun Kim,“Implementation of HTTP Live Streaming for an IP Camera using an Open Source Multimedia Converter ”, ISSN:1738-9984 IJSEIA Copyright ⓒ 2014 SERSC, pp. 39-50
[6]. R. Pantos, W. May and Apple Inc. Internet draft - https://tools.ietf.org/html/draft-pantos-http-live-streaming-20
[7]. FFMPEG, https://trac.ffmpeg.org
[8]. https://en.wikipedia.org/wiki/HTTP_Live_Streaming [07:38, 22-10-2016]
[9]. https://www.encoding.com/http-live-streaming-hls/