COMPARISON OF MODERN VIDEO STREAMING PROTOCOLS OVER HTTP: MPEG-DASH AND HLS : REPORT

PROJECT MEMBERS

- Siddhartha Mishra (ES15BTECH11018)
- Raaghav R. (ES15BTECH11021)

OBJECTIVE

The goal of the project is to implement a **Media server** compatible with both MPEG Dynamic Adaptive Streaming over HTTP (DASH) and HTTP Live streaming(HLS).

PROCEDURE:

The structure for both HLS and DASH is :

For both the methods the common structure for the Media server/client is :

Server :

- Segment the media or streaming buffer to make a playlist using ffmpeg[3] (Handle audio and video stream appropriately).
- Add appropriate signature and track information and append the parsable format as mentioned in RFC [1] for HLS and RFC [2] for DASH.
- Encode and encrypt the segments (END-to-END) and send securely over HTTP.

Client :

- Decrpyt the segments upon receiving them (Handle audio and video stream appropriately).
- Parse the receiving segments and use header information to recreate the proper sequence and adapt to the bitrate according to throughput and request different segments to server.
- Decode the sequence and recreate the playlist.
- Read from the buffer and stream it using openCV [4].

Other:

1. Also implement protocol specific logging, adaptive features etc.
2. Evaluate performance metrics to decide which method is better with respect to which parameters.

PARTS WE WORKED ON

Siddhartha Mishra

- DASH : Implementation of the segmentation, encryption and specific features and working prototype as mentioned in RFC.
- Client side : Parsing segment and decoding them to recreate playlists on client side.

Raaghav R.

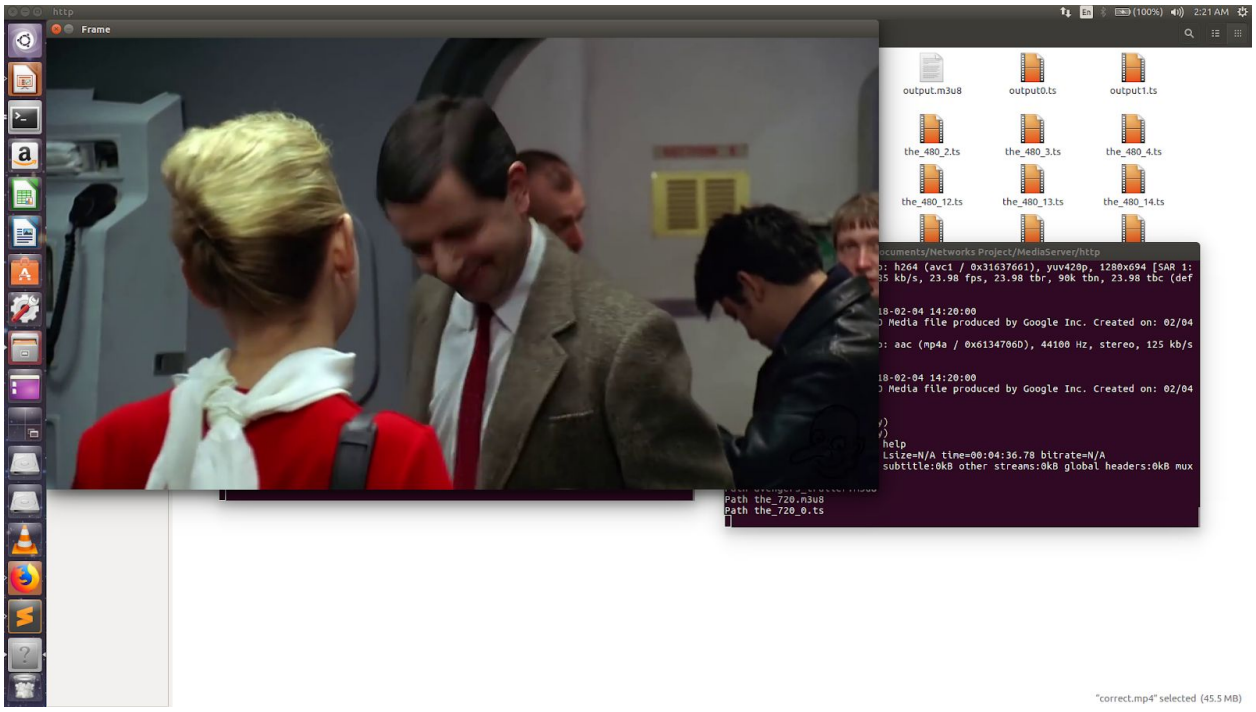
- HLS : Implementation of the segmentation, encryption and specific features and working prototype as mentioned in RFC.
- Client side : Render streaming buffer as received in order on display using OpenCV.

RESULTS:

1.HLS and MPEG server: Generates Playlists/MPDs according and triggers appropriate actions on those segments upon request by client for HLS/MPEG DASH respectively.

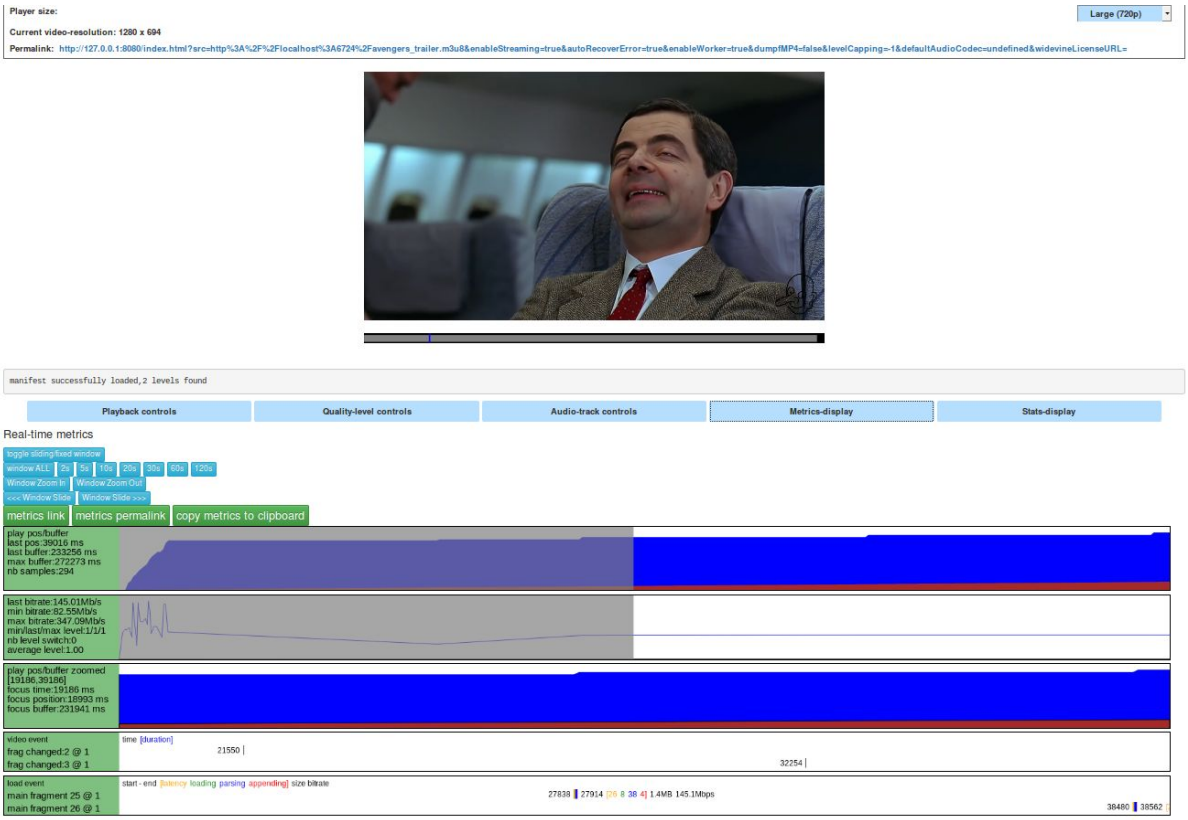
2.HLS Video Streaming Client :

Below is the working of our own custom client, that parses through the master playlist, that adapts to the current bitrate, demands the ts files needed from the quality that it can afford and plays it using OpenCV video tools.



Testing correctness of server on existing external HLS client :

This was a famous open source hls client and it our server is able to respond to it’s request and adapt to bitrates properly.



Testing correctness of server using MPEG DASH external client STATS(Dash.js) :

LIST OF TOOLS, SOFTWARES AND OTHER REFERENCES:

[1] https://tools.ietf.org/html/rfc8216(RFC for HLS)
[2] https://tools.ietf.org/html/rfc6983 (RFC for MPEG-DASH)
[3] https://ffmpeg.org Used to create ffmpeg segments of a continuous stream using a c ommon encoder so that it can be decoded properly on the client side.
[4] https://opencv.org To render video on client side stream from received buffers over network from media server.