Final Project Report

Mai Trinh
COSC 6377 Computer Networks*, University of Houston

April 29, 2021

1 Introduction & Problem Statement

Video streaming over the Internet is an important source of communication and entertainment. Video traffic from content delivery networks occupied the majority of the Internet traffic. With the large scale of video streaming applications available, the consumer demand for better quality videos is also increasing. Quality of Experience (QoE) in video streaming, i.e. user satisfaction level, is an important factor in determining the video quality. There are many influencing factors for QoE including human factor such as user's background and interest or video content and background. However, the main focus of this project is to understand the relationship between the network metrics and the human subjective Quality of Experience (QoE) in video streaming. In addition, HTTP-based adaptive video streaming, like MPEG-DASH, has become the standard for many video streaming applications such as Netflix and YouTube. The use of MPEG-DASH is an attempt to optimize video quality and maximize user satisfaction while streaming video. Thus, by understanding the effect of network conditions on QoE, the effort to maximize video QoE and user satisfaction can be further improved.

I briefly summarize the key contributions of this project as following:

1. I presented a model to predict QoE in video streaming over the Internet based on different network metrics. The model is a Recurrent Neural Network-based approach using Long Short-term Memory network to predict continuous QoE. The model is build using TensorFlow Keras trained and tested on LIVE-NFLX-II Subjective Video QoE Database [1].

2. I conducted a simple experiment to evaluate DASH streaming in video streaming services such as YouTube. I used Network Link Conditioner to emulate network settings by adjusting different bandwidth, delay, packet loss rate, and utilized Selenium WebDriver to automate the experiment process (from running to collecting the performance metrics).

2 Related work

The video streaming over the Internet that used HTTP-based adaptive video streaming has been a popular topic for research in the recent years. Thus, there have been extension works and detailed study on the video streaming quality. Objective models for continuous-time QoE prediction are implemented in [1], using auto-regressive neural networks (G-NARX) and recurrent neural networks (G-RNN). The prediction performance of the G-NARX aand G-RNN QoE Models yield the results of 0.267 and 0.276 in RMSE, respectively. Both models showed promising performance, however, further improvement can be made to obtain better results. In [2], the authors proposed an long short-term memory (LSTM)-QoE model, a recurrent neural network-based QoE prediction model on 4 different databases. This method produced excellent performance in QoE predictions, however it did not utilized the LIVE-NFLX-II Subjective Video QoE Database which is the newest collected dataset in a highly realistic streaming system.

3 Solution approach/ Design

This project is divided into 2 parts: QoE modeling and DASH testing. In the first part of this project, the main objective was to build a model that can predict QoE in video streaming based on different network metrics. The solution approach to complete such task is described below.

- Obtain the dataset, prepocessing the data into a format that is suitable for exploratory analysis and modeling such as remove unnecessary data, drop missing values and outliers, separate discrete features and continuous-time features.
- Perform exploratory analysis and features selection
- Proposed a Recurrent Neural Network-based approach to model the continuous QoE prediction.
- Perform training and testing on the proposed model

*Final project report

For the second part of this project, the main objective was to evaluate DASH streaming in video streaming services. To do so, I conducted experiment for DASH Testing on YouTube player under different network conditions. The design framework for this experiment is as follow. The testbed consists of a video server (public streaming services such as YouTube, Netflix or DASH player such as DASH-IF, Bitmovin or controlled server such as M-Lab server), an automate experiment controller and a network emulator. The controller automate the experiment by acting as the client and programmatically open web page and play the video clip. The controller collects the performance metrics and saves it to log files. After the experiment is completed, I collects the log files and displays the graphs with the experiment results using Excel. A network emulator updates the network environment by modifying factors such as the network bandwidth, delay and packet loss rate. Network simulation helps to understand how network conditions effect DASH performance. Since the video quality is proportional to the video bitrate, this experiment focuses on obtaining the variations of bitrate under different network conditions.

4 Implementation

This section describes the implementation process of the QoE predictor model as well as explains how to setup and conduct experiment for DASH testing on YouTube player.

4.1 QoE Predictor Model

4.1.1 Data analysis

Before building a predictive model, I first performed analysis on the dataset. In video streaming services, human subjective score serves as an important indicator for video quality, thus I analyze the database by means of the retrospective and continuous-time human subjective scores. Figure 1 depicts the relationship between human subjective score and objective features in video streaming quality. I found that the human subjective score is linearly dependent of video bitrate, and VMAF measurements. Note that, after about 820 frames, QoE scores show a steep drop since it corresponds to rebuffering event occured at around frame 820. The same observation can be said for bitrate and VMAF score; since VMAF discards the score of all rebuffered frames, the graph for VMAF measurement stops after 820 frames. While there are some outlier cases, we can generally say that video quality assessment (VMAF), video bitrate, and rebuffering event are important factors in determining QoE score.

Figure 1: Average continuous-time subjective scores, average video bitrate and average VMAF across frame

4.1.2 Feature Selection

Based on the results of the analysis above, I chose the following features as inputs for the QoE predictor model.

- Frame video quality: Video Multi-method Assessment Fusion (VMAF) can be measured using using libvmaf developed by Netflix. More details at https://github.com/Netflix/vmaf. The choice of VMAF is not restrictive, and other type of video quality assessment such as PSNR or SSIM can also be used in place of VMAF.
- Rebuffering: a per-frame Boolean variable denoting the presence of rebuffering event. QoE is greatly influenced by the occurrence of rebuffering event in the video sequence.
- Bitrate: the bitrate for each frame belonging to a segment. Note that the higher the bitrate, the higher the QoE.

4.1.3 RNN-LSTM Model

Each video in the dataset is considered a dataset, and each dataset is split into training-validation-test set at 70-10-20 accordingly. The model is trained, tested and evaluated on each video sequence. For each video, the model

is trained for 50 epochs. This process is carried out for all videos in the dataset. That is 420 test evaluations that the model has to performed. By doing this, we are ensuring that the trained model perform a fair evaluation.

Since we are working with continuous, sequential data, I proposed a Recurrent Neural Network based model to predict continuous QoE score. More specifically, I used Long short-term memory model (LSTM), a powerful and popular class of RNN that have shown to be effective in modelling continuous data. The summary of LSTM model for QoE prediction is shown in Figure 2. The network configuration consists of 2 LSTM layers, each with 50 units. I also added a Dropout layer with the dropout probability of 0.2, meaning that 20% of the layers will be dropped to prevent over-fitting. Thereafter, I added the Dense layer that specifies the output of 1 unit. The model is compiled using Adam optimization algorithm and the mean squared error loss function. Next, the model is to run on 50 epochs with a batch size of 1. Note that batch size is fixed to 1 as the LSTM network is stateful, that is the last state for each sample in a batch will be used as initial state for the next state. When batch size is set to 1, the network weights are updated after each training example, gives us a faster learning but also adds variance to the results. An important factor to know when training stateful LSTM is that we train it manually one epoch at a time and reset the state after each epoch. Thus, preserving the sequence in which the input training data was created, so we don't need to shuffle the input. The building of LSTM model is based on tutorial in [3], [4].

```
Model: "sequential_419"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_838 (LSTM)              (1, 1, 50)                10800
_________________________________________________________________
lstm_839 (LSTM)              (1, 1, 50)                20200
_________________________________________________________________
dropout_419 (Dropout)        (1, 1, 50)                0
_________________________________________________________________
dense_419 (Dense)            (1, 1, 1)                 51
=================================================================
Total params: 31,051
Trainable params: 31,051
Non-trainable params: 0
_________________________________________________________________
```

Figure 2: Summary of LSTM-QoE model

4.2 Experiment Setup for DASH Testing

To emulate different network settings, I used Network Link Conditioner, available in MacOS. I changed the experimental settings as follows for the evaluation in various network scenarios.

- Varied network bandwidth from unlimited to limited bandwidth with various values (1024, 2048, 3072 Kbps)
- Varied the network round trip delay (0, 50, 100, 300 ms)
- Varied the packet loss rate (0, 2, 8, 16, 25 %)

I performed the experiment in different scenarios, record the performance and evaluated the results. For this experiment, I selected A-Team Official Trailer video clip with a playback duration of 1:43. Obtaining the information about performance metrics or YouTube player requires special method. YouTube provides an option to displayed video playback information called "Stats for nerds" (shown in Figure 3). By default, this function is not enable. Thus, in order to automate the experiment, I used Selenium WebDriver to systematically interact with YouTube. With the WebDriver, I can open YouTube video web page, find the button on the web page to enable "Stats for nerds" and click it programmatically. I then parse the text in the information panel to obtain the performance metrics such as bitrate and buffer size. The results are recorded every half second.

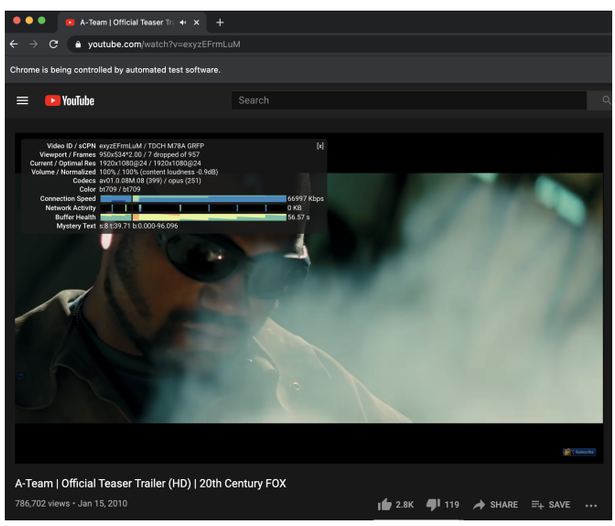

Figure 3: YouTube player with an option to display the detailed video information enabled

5 Results

5.1 Performance Evaluation of QoE Predictor

The QoE model performance is evaluated in terms of the following metrics:

- Linear Correlation Coefficient (LCC)
- Spearman Rank Order Correlation Coefficient (SROCC)
- Root Mean Squared Error (RMSE)

LCC and SROCC provides the correlation between the predicted QoE and the actual QoE. The RMSE indicates the absolute fit of the model to the data (i.e. how close the observed data points are to the model's predicted values). A good model is characterized with high LCC and SROCC (close to 1) and low RMSE (close to 0) [2]. The results of QoE prediction performance are shown in Table 1. This model shows a promising result with an average LCC score of 0.82, an average SROCC score of 0.87 and an average RMSE score of 0.10. Overall, this model performed relatively well for the given dataset.

LCC	SROCC	RMSE
0.818	0.872	0.103

Table 1: QoE Prediction Performance

5.2 Evaluation Results of DASH Testing

The continuous bitrate of video under different network setting are recorded and displayed in Figure 4. When testing under different bandwidth while keeping delay at 0 ms and packet loss rate at 0%, as expected, the video bitrate increases as the network bandwidth increases. Similar process applied when testing the impact of round trip delay on birate, the only network parameter that varied in this case is the delay with the default packet loss rate of 0% and unlimited bandwidth. The bitrate decreases when the delay increases, except for the case at 100ms delay, the average bitrate is shown higher than that of 50ms delay. We presume that the unexpected results is due to the possible background traffic that takes up most of the network throughput when the network is set with 50ms delay. Thus, more advanced scenarios can be performed to further evaluate the impact of network conditions in video streaming. Lastly, the relation of the bitrate and packet loss rate is performed as expected: as the packet loss rate increases, the bitrate decreases. This result is further clarified in Figure 5 showing the average bitrate under different bandwidth, delay and packet loss conditions.

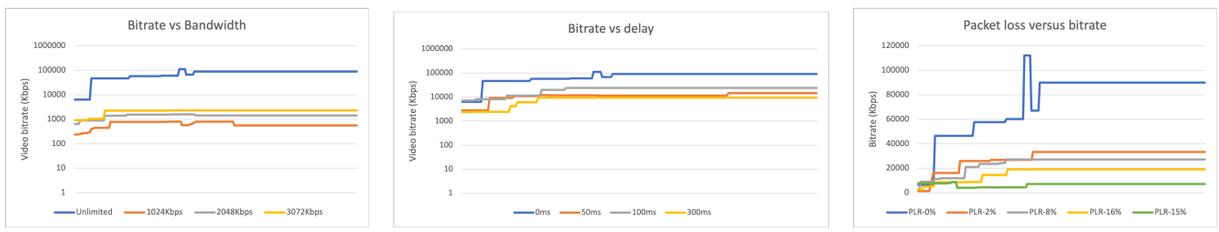

Figure 4: The continuous bitrate in different bandwidth, delay, and packet loss conditions

	1024Kbps	2048Kbps	3072Kbps	Unlimited	
Average Bitrate	627.6407767	1394.650485	2151.84466	73566.93204	
	0ms	50ms	100ms	300ms	
Average Bitrate	73566.93204	11427.60194	20395.15534	8183.76699	
	PLR-0%	PLR-2%	PLR-8%	PLR-16%	PLR-15%
Average Bitrate	73566.93204	28401.68447	23842.57767	16042.59223	6548.35922

Figure 5: The average bitrate in different bandwidth, delay, and packet loss conditions

6 Conclusions & Future work

In this project, I presented RNN-LSTM model for predicting continuous QoE in video streaming over the Internet based on network metrics. Using the LIVE-NFLX-II database, I was able to analyze continuous-time user experience under different network conditions. I also trained ad evaluated predictors of quality of experience on this dataset. The overall QoE prediction performance showed that the designed RNN-LSTM shows a reasonably good result for the provided data. In addition, I also performed an evaluation of video streaming services (YouTube) developed under the MPEG-DASH standard. The experiment was conducted under different network conditions and made

observations on the performances of YouTube player under simple scenarios. Overall, the experiment shows that MPEG-DASH technique used by YouTube player performed as expect for simple network variations, with the exception of scenarios where results are impacted by background traffic.

Note that there are limitations to this project. Particularly, LIVE-NFLX-II Video QoE database have limited features thus the presented QoE model does not reflect the real network environment. By integrating additional features such as resolution changes, buffer status, network throughput, the proposed model can be modified to achieve the good performance under a highly realistic streaming environment. For future work, this proposed QpR predictor model can be integrated into the client-adaptation algorithm to optimize video streaming quality of experience. For the second part of this project, the experiment environment are confined to local network rather than the mobile, wireless network. The experiment limited to only testing YouTube player, but other streaming services can also be tested under the same setting. Additionally, more advanced scenarios can be performed to further evaluate the impact of network conditions in video streaming, such as the impact of multiple players, background traffic, and dynamic bandwidth.

Appendices

Final Project Proposal Checklist

	Tasks	**Status**	**Note**
Milestone 1	Perform analysis on the data	Completed	
	Features selection for the model	Completed	
	Build a QoE predictor model based on the selected features	Completed	
Milestone 2	Perform evaluation of the proposed model	Completed	
Additional implementation	Simulate a video streaming session over the network and apply the algorithm into client-adaptation strategy to optimize video streaming	Incomplete	After discussing with Dr.Gnawali, I decided to change this additional implementation into performing DASH Testing.
	Evaluate DASH Streaming	Completed	

References

[1] Christos G. Bampis et al. *Towards Perceptually Optimized End-to-end Adaptive Video Streaming*. 2018. arXiv: 1808.03898 [eess.IV].

[2] Nagabhushan Eswara et al. "Streaming Video QoE Modeling and Prediction: A Long Short-Term Memory Approach". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.3 (2020), pp. 661–673. DOI: 10.1109/TCSVT.2019.2895223.

[3] *Stateful and Stateless LSTM for Time Series Forecasting with Python*. URL: https://machinelearningmastery.com/stateful-stateless-lstm-time-series-forecasting-python/.

[4] *Using a Keras Long Short-Term Memory (LSTM) Model to Predict Stock Prices*. URL: https://www.kdnuggets.com/2018/11/keras-long-short-term-memory-lstm-model-predict-stock-prices.html.