Article

Quality of Experience Experimentation Prediction Framework Through Programmable Network Management

Ahmed O. B. Al-Mashhadani [1,†,‡], Mu Mu [1,‡] and Ali Al-Sharbaz [2,*]

[1] Faculty of Arts, Science and Technology at the University of Northampton; research@ahmedosamabasil.com
[2] University of Gloucestershire; aalsherbaz@glos.ac.uk
* Correspondence: e-mail@e-mail.com; Tel.: (optional; include country code; if there are multiple corresponding authors, add author initials) +xx-xxxx-xxx-xxxx (F.L.)
† Current address: Affiliation 3
‡ These authors contributed equally to this work.

Abstract: Quality of Experience (QoE) metrics can be used to assess user perception and satisfaction in data services applications delivered over the Internet. End-to-end metrics are formed because QoE is dependent on both the users' perception and the service used. Traditionally, network optimization has focused on improving network properties such as the QoS. In this paper we examine the Adaptive streaming over a software defined network environment. We aimed to evaluate and study the media streams, aspects affecting the stream, and network. This was done to eventually reach a stage of analysing the network's features and their direct relationship with the perceived QoE. We then use machine learning to build a prediction model based on subjective user experiments. This will help to eliminate future physical experiments and automate the process of predicting QoE.

Keywords: QoE; Fairness; SDN; Classification Prediction; DASH; Multimedia

1. Introduction

MPEG Dynamic adaptive streaming over HTTP (DASH) is an adaptive bit-rate streaming methodology which has many abilities providing best quality streaming of multimedia related applications across the internet progression from traditional HTTP web servers. It works on the criteria of breaking down video content into small sequential data segments, which further worked over HTTP. Every segment has a small time duration of playback, that consists of multiple characteristics; like a short movie clip or the time broadcast of an event-like show or sports program. The MPEG-DASH can adapt to alternating network fluctuations and enable the best quality playback with a minimal number of re-buffering occasions. One of the main problems that we are trying to solve with our research is figuring out the correct classification methods for noisy data such as the output of a DASH media stream.

There is a considerable rise in the general quality of experience (QoE) anticipated by the users across multimedia distribution strategies like live streaming of video. Since the number of online client implementations and the equipment usages increased, there is a significance change in the capability of the end equipment and facilities such as network bandwidth, which are usually distributed among various end user equipment. Traditionally best-effort network construction assigns assets depending upon request of client and advance-level Service Level Agreement (SLA) without putting application and user level necessities into consideration. The end-users are usually unsatisfied due to the perceivable unfairness. The concept of fairness within a network or between network resources can be interrupted in many ways and thus fairness can also be achieved in many different ways depending on the scenario [1].

In this paper, a new quality analysis method is used as a distant broadcast technique for measuring the appropriateness of the environment for contributing in multimedia

 doi:10.20944/preprints202207.0035.v1

video evaluation. In a laboratory experiment, contributors (past researchers) achieved this quality analysis with various listening devices in various listening surroundings, involving a silent room allowing a imitation circumstantial noise situation. Results show important observations of the situation and the attending device on the quality inception. Thus, the arrangement of our video trials will be free of sound and only target one aspect, the quality of the video perceived. We aim to tackle the issues of subjective evaluation of objective QoE models and Adaptive Bitrate Algorithms (ABR). We proposed an experimentation framework structure through programmable network management for the generation of Machine Learning (ML) Training-Ready Data and MOS/QoE Prediction. We used our testbed's data-generated analysis with real user experimentation and ML for training and predicting QoE based on the generated monitoring data. This way with the generated prediction, limited user-testing is needed in the future, hence future researchers will simply run the generated monitored data from the network tracing tools into the prediction model and it will generate a predicted MOS for faster and more efficient network level tests and experiments. This model is unique because we tested its data with state of art ML algorithms and achieved highly accurate prediction results. The main contributions and findings of this work outline,

- A Virtual-Box Environment with all the necessary libraries and applications needed to run P4 [2], Openflow [3], Python 2 and 3 instances, DASH and Mininet. With all essential packages installed, an error-free test environment.
- A Segmented content database with 6 source videos, 120 test videos, H.264 encoding configuration at 6 levels and is resolution adaptive with full configurable options. This dataset was generated to include as many encodings, capabilities and features into the model so that it may be useful in countless situations.[4]
- A P4 SDN testbed over Mininet with the ability to control DASH initial buffering, stalling, switching, monitoring, bitrate adaptation, and bandwidth limitation over selected ports. The testbed provides the capacity for comprehensive uesr experiments and data collections, which lead to our insights and analysis of congestion for congestion related experiments. Along with full re-configurablity over data plane and everything mentioned above.
- A proposed experimentation framework structure through programmable network management for the generation of ML Training-Ready Data and MOS/QoE Prediction.
- Human experiment with QoE MOS-based feedback to benchmark the accuracy of predicted QoE and Network Features.
- Analysis of state of the art machine learning algorithms, along with the creation of an experimentation framework for feature evaluation in network experiments.[4]

Database	Source Videos	Test Videos	Encoding Configurations	Test case Formation	HAS-related Impairments	Resolution Adaption
LIVEMVQA [5]	10	200	H.264 at 4 levels	hand-crafted	switching or stalling	No
LIVEQHVS [6]	3	15	H.264 at 21 levels	hand-crafted	switching	No
LIVEMSV [7]	24	176	no compression	hand-crafted	stalling	No
Waterloo SQoE-I [8]	20	180	H.264 at 7 levels	hand-crafted	switching	Yes
LIVE-Netflix Video QoE Database [9]	14	112	H.264 at 6 levels	hand-crafted	initial buffering & Stalling & Switching	No
Waterloo SQoE-III [10]	20	450	H.264 at 11 levels	simulated	initial buffering & stalling & Switching	Yes
ITEC DASH [11]	7	131	H.264 at 6 levels	hand-crafted	initial buffering & stalling & switching	Yes
Our Dataset	6	120	H.264 at 6 levels	simulated	initial buffering & Stalling & switching & Monitoring	Yes

Table 1. Comparison of publicly available QoE Dataset for HTTP-Based Adaptive Video Streaming

2. Problem Space & Related Work

HTTP adaptive streaming (HAS), such as MPEG-DASH splits a broadcast file into many segments. Every segment is encrypted into number of bitrates to attain the access for user with changing stream specifications. The video segments are consecutively requested by the client, having maximum bitrate possible, which is approximated based on the network and user needs. The process of ABR is through which a client suggests the ideal bitrate of part to download. ABR contains default conditions, but many of them do not reflect the difference between the multiple scenarios that may occur in a production environment and often work poorly when the organization makes changes in the work environment. A recent attempt is to link the increase in ABR to the capabilities of the interpolation method [12], which is mainly based on a machine learning model. The ML-based ABR strategy is divided into two parts. In the first part, you can adjust the current ABR parameters. A calculation plan based on ABR variables is proposed. Support systems where ABR can change variables depending on the order in which the network conditions are changed. In deep learning, this ABR variable is based on an adaptation policy, where changes are presented as the context of the flow. In deep learning, depending on the strategy used in which tuning of the parameters of ABRs, would directly affect the streaming content.

Previous research states that bitrate depends upon ABR models having trained predictive collection of decisions (SMASH [13]). Where a 'combine grouping' scheme was used to make a map network-related properties of bitrate. A supervised ML-based ABR was implemented with features related only to the bitrate status. Therefore, there are some restrictions with those predictions as features are focused on limited network factors. Both the conception of engineering and the ML algorithm selections will not be executed in a methodical fashion, moreover the trained model has planned to support, rather than replace, the existing fixed-rules that depend upon modifying the algorithm. Moreover, recently introduced model-free strategy is said to be Pensieve [14], which uses reinforcement learning strategy to introduce a neural network based upon ABR. This strategy makes no explicit assumption related to effective data. Multiple papers have reported issues like Pensieve, therefore it is being accompanied by implementing detailed experimental evaluation of Pensieve, keeping various sets of video content and network go over under observation [15]. In [16] within the training process, the results change significantly when

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

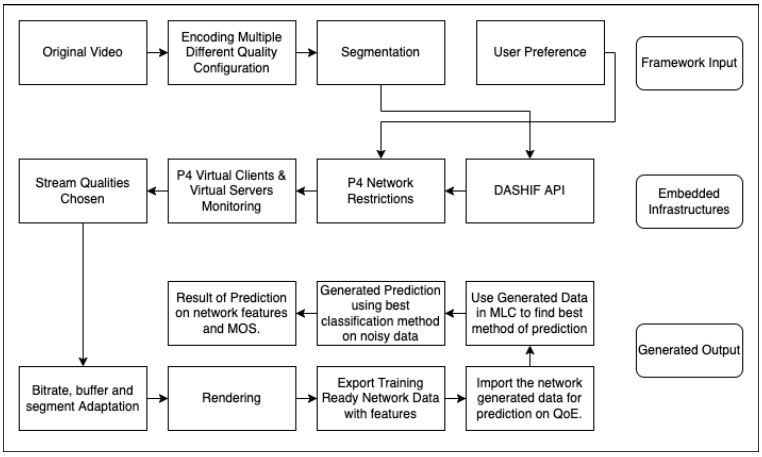

Figure 1. Proposed Experimental Framework for Feature Evaluation

using a web account, the deficit has a high value and does not tend to coincide. They have a high bitrate presented in their video setup. For example, when running UHD and 4K data packets, authors choose reasonably bright screens based on past learning success, indicating that the data is fragmented. A bitrate equal to the brightness percentage is reduced by 50%, the experimental model only learns to achieve maximum results, and the bitrate leads to an inaccessible video level. The heterogeneity of wireless networks means that larger variants continue to be known, and as the value of video resolution continues to grow, so does the popularity of data. This work was therefore encouraged. For research and experimentation, researchers spend an enormous amount of time in the creation of their test environment, most of these environments are quite specific to the target of their research. There are many downloadable virtual environments such as, P4 or SDN based virtual testbeds however, these environments are very specific to their purpose. Configuring a suitable testbed that has the ability to run multiple solutions from multiple different testbeds is very time consuming. Thus we created a Virtual-Box Environment with all the necessary libraries and applications needed to run P4, Openflow, Python 2 and 3 instances, DASH and Mininet. This provides ease for researchers to use our virtual machine setup to dive straight into testing and data generation without wasting their time and effort building the virtual setup.

Table 1 shows a list of previous HTTP Adaptive Video Streaming Databases that are widely used in research within QoE. Our generated Database of encoded videos contains 6 source segment division videos and 120 different resolution videos. Our database comes with a pre-configured P4 Software Defined Network with a server and multiple clients for testing and evaluating QoE with DASH Reference Player. Its main contribution is that it monitors the network and all its ports for recording DASHIF Reference Server packets and evaluating client's data. It then generates a training ready clean data for ML usage purposes. This was done to ease the experimentation with big data and data mining for use in ML and the understanding of multimedia network environment.

Furthermore QoE is widely discussed and predicted in multiple aspects within the use of different features and multiple joint classifiers [17–21]. Our approach is unique in its data generation. We depend on our testbed to generate the network data required, and based on that network data, we use machine learning algorithms to predict the generation of MOS and all other features, in addition to this, we analyse the most effective feature from the network data that directly affects the MOS and prioritize it in our testbed. Thus the data generated from the test bed is always training-ready to test on more algorithms.

3. Preliminaries & Methodology

The Approach presented in this paper makes use of different configurations of neural network and classification arranged to provide the best fit feature classifier and MOS

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

prediction suitable for most of the tasks characterizing modern media streaming which is the specific goal of this work. Moreover, we focused on the design and implementation of our P4 testbed to host the DASH Reference player and ability to monitor it, where a researcher can easily extract the data and use the ML techniques we show in this paper to test and improve upon this work.

3.1. Adaptive Streaming

Multimedia data file is segregated into multiple parts or segments and then conveyed to the user using HTTP. A media presentation description (MPD) explains the particulars of the segment, the particulars of the segments include aspects like time, website, multimedia properties such as video resolution and bit rates. These segments can be arranged in number of methods like segment base, segment timeLine, segment template and segment list, depending on the use-case. Segment can be media file of any type or format, such as the "ISO base media file format" and MPEG-2 Transport Stream; there are the two major kinds of container's format. DASH can be considered as a Video/Audio codec sceptics. Media files are usually provided as a multiple number of illustrations and the concerned choice of data is mainly related to the network status, equipment potential and client preferences that are responsible for allowing the adaptive bitrate streaming and impartiality of the Quality of the Experience.

The adaptive bitrate streaming logic is not defined by MPEG DASH standard. Therefore, DASH can be implemented on any type of protocol. PCC expressively decreases the storage quantity at the cost of multifaceted pre-processing and execution at the client. "HTTP adaptive streaming" (HAS) deals with dynamic setup circumstances while endeavouring transport at maximum quality possible in the given circumstances.

3.2. DASH Objective Metrics

The basic remodelling of the automation is the media streaming that includes best quality on demand data and live media content. In the present scenarios the main interest is to attain the pre-eminent quality of service and experience because of the ever aggregating network consumption and the user demand. Traditional type of streaming methodologies confront multiple trials in distributing multimedia content to the end user without lowering the quality of the service. The Adaptive HTTP streaming is the ever increasing content providing tactic; which delivered the real time content without negotiating quality and guaranteed the excellent quality of experience. The selection of bit rate must be effective and durable, it depends upon the nature of the network and thus we argue that it must always be dynamic. The client key potential point outs towards gaining the outstanding quality of service towards the end user, to achieve an increase in the range of standards and procedures introduced in the Adaptive HTTP streaming field. Researching and contrasting is compulsory for executing numerous techniques depending upon predefined merits. The HTTP Live Streaming, Microsoft Smooth Streaming and MPEG DASH are arising model techniques of adaptive HTTP streaming. In order to calculate the transporting execution, the experiment moves forward using G-streamer adaptive HTTP streaming. In order to calculate the transporting execution for achievement; is based upon changing multiple networks and situations for on-demand streaming and live streaming content. The adaptive HTTP streaming method's resultant is calculated and examined using pre-explained performance indices. The processed data depicts that each entertained method of delivery and best conductance is achieved by the predefined advantages. In short, DASH provides appreciable balanced performance throughout multiple networks arrangements as compared to other streaming approaches.

In this paper, the experiment is staged based on the encoding information in Table 2. User will watch a video based on the video segmentation and resolution properties mentioned in Table 2, all videos are fixed on 30 frames per seconds on all video streaming qualities, with a minimum buffer time of 2 seconds of loaded content. User will watch five 40 seconds trials of video content on a limited bandwidth of 0.5 mbps, 1 mbps, 3 mbps, 5

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

Codec	Bandwidth of Activation	Resolution
avc1.64001f	3134488 bps	1024x576
avc1.64001f	4952892 bps	1280x720
avc1.640028	9914554 bps	1920x1080
avc1.64000d	507246 bps	320x180
avc1.640015	759798 bps	480x270
avc1.64001e	1013310 bps	640x360
avc1.64001e	1883700 bps	768x432
avc1.640033	14931538 bps	3840x2160

Table 2. Video Database Encoding Information

mbps, and unlimited settings. With every trial, user will input a Video MOS (vMOS) rating based on their video experience of initial load delay, resolution change, and over all quality of experience.

3.3. Subjective Evaluation of DASH

In principle, the aim is to evaluate the relevant QoE parameters and variables that take into account the kinetic properties of the video. Common objective indicators of a subject's performance to regularly determine their relevance to human perception are essential for evaluation. There is no subjective evaluation for DASH Adaptive streaming, for justifying longer video patterns; which are enough to explain the birate switching for the data-set that acquires the longer segment videos to various network conditions sequences. Estimation of the end user's real-time Quality of Experience (QOE) online by exploring the apparent influence of delay, diverse packet loss rates, unstable bandwidth, and the apparent quality of using the altered size of DASH video stream segment over a video streaming assembly under multiple video arrangements is quite possible with this paper's approach. The performance and potential of the system and prospects of the end user depends upon the Mean Opinion Score. The subjective evaluation of DASH gives an overview of impairments with various network and various video segments on different end-users. For the test setup and procedure, we used the most recent ITU-T P.913 Titled: "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment." [22] as general guidelines. A screen with 4K resolution was used to passively stream the content and record its segmentation. Which was used to compile as a separate video and present to the user to eliminate Virtual Environment computational power limitations from affecting the video and its MOS rating, we recommended sitting at a distance of approximately four times the height of the screen used by users. The applied test protocol was as follows: Firstly we started with the Welcome text based information; Briefing and informed consent. Then we moved on to the explanation and recommendations of the Setup as recommended by the ITU-T; Screening and demographic information. Following that, we showed our content, 5 video samples based on different networking runs. Our Evaluation stage was next, which was a collection of 3 QoE related questionnaires for each video sample. Ending with the Debriefing which entitled the Feedback and remarks from user-end. This way we have the MOSes from the users and the recorded network data of their experiment to use for our prediction mechanisms later. Figure 2 defines MOS where R represents the user's ratings for the given question and the question's representation of the video's stimulus. The MOS ratings were defined into 5 different classes before uploading the rating data to the training process as shown in Table 3.

4. User Experimentation

In our previous paper [23], we discussed the creation of the user experiment, in this paper, we provide a detailed perspective of the experimentation that was created.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

$$MOS = \frac{\sum_{n=1}^{N} R_n}{N}$$

Figure 2. Where R are the individual ratings for a given stimulus by N subjects.

MOS	Definition	Description	Class
1	Bad	Unsatisfactory Perceived Quality	1
2	Poor	Unsatisfactory Perceived Quality	2
3	Fair	Acceptable Perceived Quality	3
4	Good	Satisfactory Perceived Quality	4
5	Excellent	Highly Satisfactory Perceived Quality	5

Table 3. Mean Opinion Score Scale

After careful consideration of the time, place and settings, we followed the ITU guide for experimentation, thus based on that guide, this section explains the user experimentation process and test-bed used. ITU-T P.913 Titled: "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment." [22] was ideal to use due to the fact that we were in an epidemic situation. The guide reflected accurate methods of experimentation in "any environment" so we designed the experiment remotely with control over the main key technological aspects.

4.1. General Viewing Conditions

The viewing conditions of this experiment will be evaluated based on the user and their screen. The viewing distance is chosen to be the preferred viewing distance (PVD) which is based upon viewers' preferences. These are the recommendations used for the experiment. Due to the fact that the experiment took place on a remote non-monitored platform, the user is informed before downloading the sample video to change their screen settings to the most default settings and ITU recommendation settings, and must state the quality feedback of their screen, this way we have enough information to choose one set of static screen quality and its default options. The user then will evaluate their experience in the form of MOS 5-point scoring system. The MOS ratings will help us identify the user's preference towards the video and its settings shown in Table 4.

4.2. Technical Testbed Setup

Our test-bed is built on an Ubuntu machine running multiple items. The SDN environment was set-up on Mininet virtualisation platform. The network backend was programmed with P4 Language, we extended the basic L3 forwarding with a scaled-down

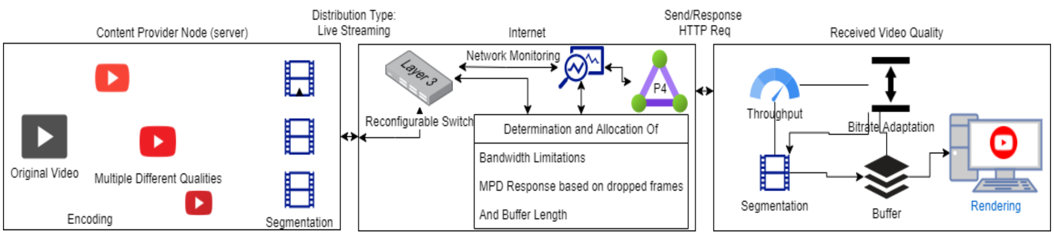

Figure 3. Testbed Overview

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

Bandwidth	Anchor	Initial Loading Time	Quality Switch Pattern
Low	Low Quality Reference	Long	Auto but constant low quality allocated
Medium	Medium Quality Reference	Short, but noticeable	Auto but constant mid range quality allocated
High	High Quality Reference	Very Short	Auto, but maximum ranges of quality are allocated

Table 4. Test Conditions (based on ITU-T recommendation)

version of In-Band Network Telemetry (INT), to make it simpler to have a multi-hop route inspection. This configuration was done to make the test-bed have universal application of testing multiple virtual devices and or a single device and a server. This configuration allowed the developer to track the path and the length of queues that every packet travels through. Multiple issues with normal Openflow configuration arise at this stage [24], thus the programability of data plane with P4 helped to append an ID and queue length to the header stack of every packet. At the destination, the sequence of switch IDs correspond to the path, and each ID is followed by the queue length of the port at switch. With this a developer will need to define the control plane rules as done with any Openflow application (but with P4) and on top of that we must implement the data plane logic of the P4 controller. This will give the user the ability to not only monitor one aspect of the network, but all ports, identifying multiple monitoring applications such as congestion which was discussed in one of our previous papers [1]. Furthermore we created a DASHIF Reference Player Server node on our test bed, and a client host from another part of the network where the client streams the video segments of the DASH server, through the network we monitor all routes and save all the network data and the reference's broadcasted data to experiment with in machine learning. There were multiple reasons why we avoided the user-testing to happen on our virtual platform apart from the global COVID-19 pandemic, after short testing we realised a noticeable latency delay that was not recorded by the network monitoring techniques that we implemented. This was due to the limitation of the virtual machine, while rendering a video we realised that the machine's CPU was over-exhausted. Thus the user MOS rating will be affected by non-network factors which we wanted to eliminate for accurate results by converting the recorded segments into an mp4 file to be downloaded and run by the tester. Figure 3 shows the process of the DASH player live streaming to a user over HTTP send and receive requests and the adaptation of video quality. Figure 4 shows the experimentation process from technical server and client ends. All switches were assigned and defined with IP address and port numbers known to the development side of the process for data monitoring, moreover links were assigned experimental bandwidth limitations on the client's end to understand the patterns of network flow from the server node and to use them later as support experiments for QoE classification and prediction. Our generated data, even though it was one type of video content (Animation video), has network traces that help us in the creation of multiple predictors and the ability to compare and contrast them to choose the best fit for all future video data. [23] Table 5 shows the experimentation video map and test conditions.

doi:10.20944/preprints202207.0035.v1

Video	Controlled Bandwidth Limitations	Observed Quality Range	Observed Initial Load Delay	Configuration of Quality Switch Pattern	Resolutions Chosen from Segmentation Collection
Video 1	Limited to 0.05 Mbits/s	45373	2.66 s (Long)	Auto but constant low/bad quality allocated	1 out of 20 Resolutions Chosen
Video 2	Limited to 0.1 Mbits/s	45373 to 88482	2.2 s (Long)	Auto but constant low/poor quality allocated	2 out of 20 Resolutions Chosen
Video 3	Limited to 0.3 Mbits/s	45373 to 317328	1.58 s (Short but noticeable)	Auto but constant mid range quality allocated	2 out of 20 Resolutions Chosen
Video 4	Limited to 0.5 Mbits/s	45373 to 503270	1.52 s (Short but noticeable)	Auto but high ranges of quality allocated	3 out of 20 Resolutions Chosen
Video 5	Unlimited	987061 to 3792491	1.17 s (Very Short)	Auto but maximum ranges of quality allocated	2 out of 20 Resolutions Chosen

Table 5. Experimentation Video Map and Test Conditions

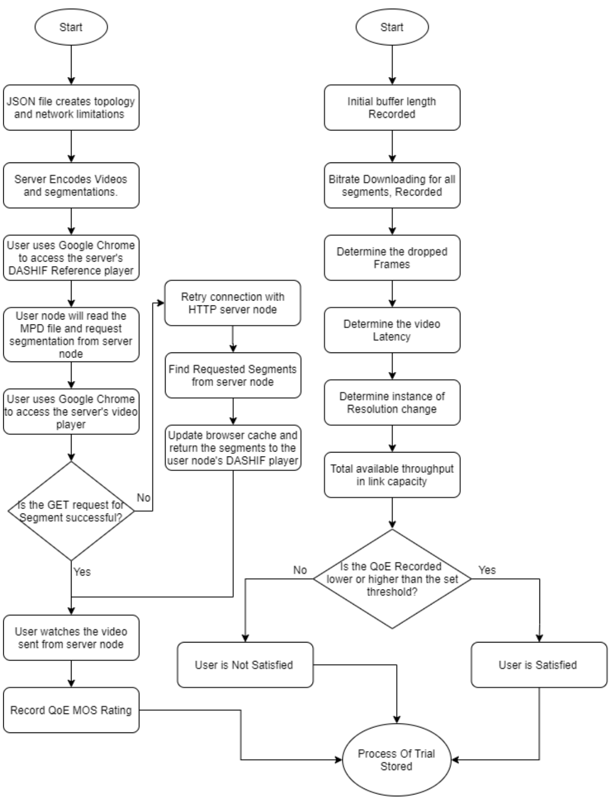

Figure 4. System Testbed Process and QoE Evaluation

 doi:10.20944/preprints202207.0035.v1

Dataset	Features
QoE Recored Parameters	Initial Buffer Length
	Live Buffer Length
	Bitrate Downloading
	Dropped Frames
	Latency
	Round-trip Time
	Video Resolution
Scoring Factors	vMOS

Table 6. Generated Dataset

5. Discussion Data Analysis

5.1. Experiment Plan & Data Description

In this experiment we use DASHIF Reference Player Web Streaming Application for adaptive streaming to users. This is to collect stream properties, understand user response based on quantitative research. Firstly the streaming server will stream selected videos in fixed properties such as a collection of fixed frame-rates, and resolutions based on chosen video segmentation. Fixed network properties to control the experiment from a network back-end perspective, on the web form there will be a MOS rating where the user shares his/her experience. The quantitative questionnaires will be limited to a rating from 1 to 5 to reflect on the relevant chosen video. All network data is being captured from user including the questionnaire, screen properties and stream. In addition to server-side streaming properties, data recorded will be placed in the processing phase, then we will conclude based on highly impacting features whether a user is satisfied or not. The experiment's expectation should outline the correct parameters (such as the manipulation of resolution) that will be used in the next experiment as an editable user choice configuration. A realistic video MPD data-set was generated based on segment collection. This data-set represents a group of MPD manifests, and m4s, encoded to run on DASHIF reference player for dynamic adaptive streaming testing. We then extract from the manifests 7 video features that will be our main input to the data training and these include; initial buffer Length, live buffer length, bitrate downloading, dropped frames, latency, and video resolution with indexing information and three MOS rating made by every user while they are streaming as shown in Table 6.

Furthermore our training data consisted of multiple properties that made them precise and unique, for the lowest network limitations that was implemented the downloading bitrates ranged from 45373 to 88482 bps with a noticeable initial loading delay that averaged out to be around 2.4 seconds. Medium networking runs provided a better range of downloading bitrate averaging around 317328 bps and 1.58 second of initial load delay. Finally when the network was given a bit more space and room to work with, the data showed an approximate bitrate download of 2147880 bps and a 1.3 seconds of initial loading delay. This is expressed in Table 7 for a clearer view. It is also important to know that the testbed was engineered to monitor the affects of network congestion on the quality of experience. This study shows the process of training the output data first to understand and generate a predicted MOS based on the trained model, increasing the ease of analysis of applications and tests that require MOS prediction within the testbed without the need for more human ratings.

Figures 5, 6, 7, and 8 show the bitrate downloading, resolution, dropped frames, and buffer length. Figure?? shows the vMOS of the users on a 3D scatter plot where the X-Axis shows the time frame, Y-Axis showing the users who participated, and finally Z-Axis with the MOS ratings, this figure explains the number of recorded responses and we can see the direct relation of the network features to the MOS and the time frame of ascension. It

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

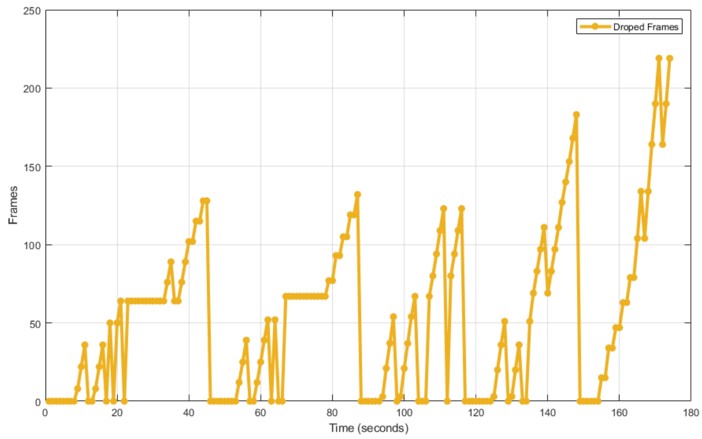

Figure 5. Results of Dropped Frames network feature experimentation trials

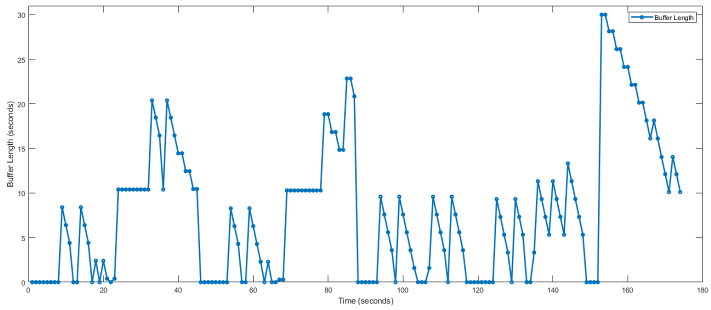

Figure 6. Results of Buffer Length network feature experimentation trials

seems that the data presents a logical front of an increase in performance that directly leads to an increase in vMOS for the quality preserved by the users.

5.2. Machine Learning Classification

Furthermore a data analysing process had to happen on two different levels post the obtaining of raw data from the virtual testbed and the vMOS from the users. QoE scoring factors and network data had to go through data cleaning and normalisation. Then the features stated before were placed through training process against the MOS and vice versa to make a collection of classification predictions and place through a neural network to compare and contrast. All features were trained with Fine Tree, Medium Tree, Coarse Tree, Kernel Naive Bayes, Linear SVM, Quadratic SVM, Cubic SVM, Fine Gaussian SVM,

Video/ Range (Mbits/s)	Quality Range (bps)	Initial Load Delay (Sec)
0.05	=<45373	2.66
0.1	=<88482	2.2
0.3	=<317328	1.58
0.5	=<503270	1.52
unlimited	=<3792491	1.17

Table 7. Experiment Ranges

 doi:10.20944/preprints202207.0035.v1

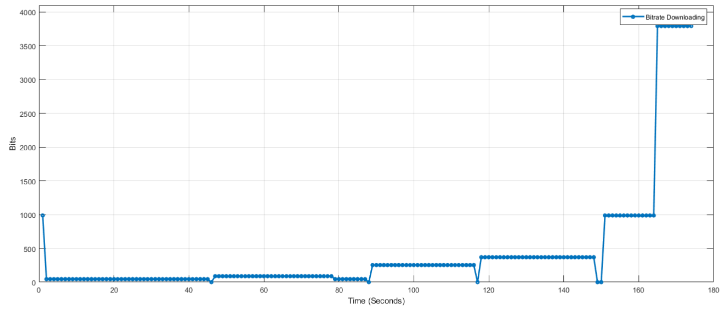

Figure 7. Results of Bitrate Downloading network feature experimentation trials

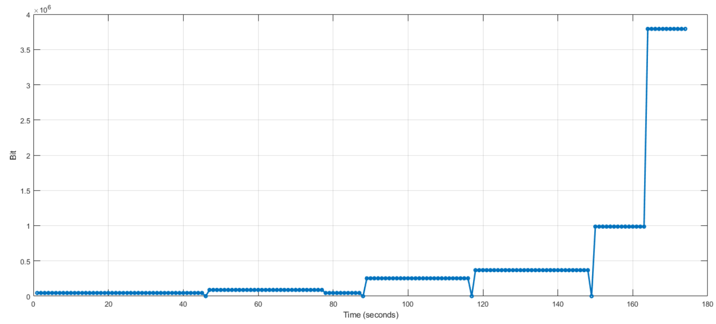

Figure 8. Results of Resolution network feature experimentation trials

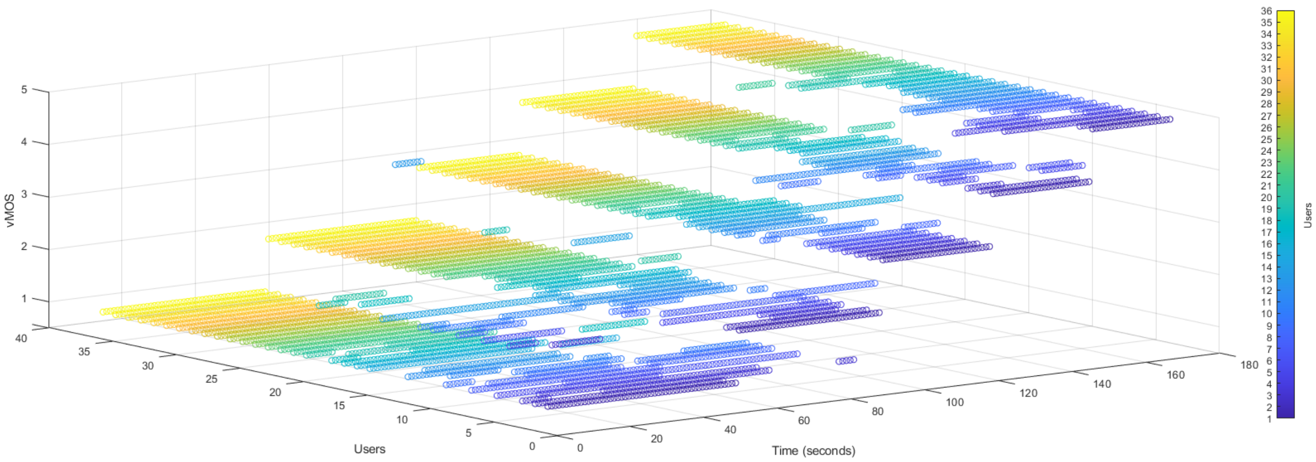

Figure 9. Results of vMOS experimentation trials

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

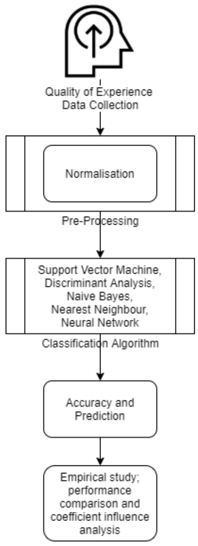

Figure 10. Data Analysis

Medium Gaussian SVM, Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Cubic KNN, Weighted KNN, Boosted Trees, Bagged Trees, Subspace Discriminant, Subspace KNN, and RUSBoosted Trees. With no over-trained attempts, this way it can be compared and contrasted which model is best fitted for the data and the feature classified [23]. Figure 10 shows an example of the data analysis process.

In Table 8, all classification methods mentioned above are used to classify the 5 features and predict their outcome if a new stream of data is inserted. The table shows the percentage of the predicted class against true class. From these models, the researcher selected the models with the highest accuracy along with the fastest prediction time. Bagged Trees was selected for bitrate downloading and MOS, Fine KNN for the Buffer Length and the Dropped Frames, Fine KNN was selected for the Resolution. Table 9 shows the most ideal machine learning classification prediction methods on highly noisy data such as our generated network data and MOS user feedback technique proposed in our previous paper [23].

5.3. Models Description & Neural Network

As described earlier, for each architecture we tried different hyper-parameters to see the best fit for the proposal. More precisely, we varied the following; 1. the use of cross-validation folds to protect against over fitting by partitioning the data set into folds and estimating accuracy on each fold (from 2 to 50 folds). 2. Number and types of splits (diversity indexes, surrogate decision splits). 3. Training Algorithm. 4. Number of layers in a network. 5. Number of neurons for a layer. 30% of our data was randomly taken out of the training phase to use for validation. Ultimately we considered the classification of all features and their classes as shown in Table 8 and the use of Neural Network on MOS prediction using the Bayesian Regularization with 2 layers, 10 neurons, using Mean Squared Error for performance rating. This resulted in 0.999 regression value which means that the correlation between the output and the target is highly accurate. So after comparing and contrasting of the classification metrics used, we recommend the mentioned models above for each one of the features as an accurate prediction method. The study shows that these models are quite accurate when it comes to DASH related streaming data and real user MOS. Neural Network method with the selected options stated above tend to be more accurate towards small or noisy data sets. However takes more time and computing power and will be inefficient to be placed within a testbed for auto prediction and redirection of resources. For such noisy data, prediction methods must be tested and uniquely selected for the prediction of each and every feature that is dependant on all other features, Figure

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 July 2022 doi:10.20944/preprints202207.0035.v1

Feature/ MLC Prediction	Resolution (%)	Buffer Length (%)	Bitrate (%)	Dropped Frames (%)	MOS (%)
Fine Tree	**98.8**	41.2	**97.6**	47.1	99.4
Medium Tree	**98.8**	42.4	**97.6**	47.1	99.4
Coarse Tree	92.9	41.2	84.1	40.0	99.4
Kernel Naive Bayes	97.6	N.A.	N.A.	60.6	99.4
Linear SVM	74.1	44.1	66.5	42.9	79.4
Quadratic SVM	78.2	88.2	70.0	82.4	79.4
Cubic SVM	78.2	88.2	70.0	86.5	79.4
Fine Gaussian SVM	77.6	38.8	65.3	77.6	79.4
Medium Gaussian SVM	78.2	38.8	65.9	46.5	79.4
Coarse Gaussian SVM	70.6	39.4	55.9	44.7	79.4
Fine KNN	78.2	**91.2**	78.2	**91.8**	79.4
Medium KNN	75.9	40.0	78.2	41.2	79.4
Coarse KNN	32.4	28.2	31.2	28.2	26.5
Cosine KNN	75.9	40.0	78.2	41.2	79.4
Cubic KNN	75.9	40.0	78.2	41.2	79.4
Weighted KNN	78.2	**91.2**	78.2	**91.8**	79.4
Boosted Trees	32.4	48.8	96.5	61.8	26.5
Bagged Trees	97.1	36.5	**97.6**	39.4	**99.9**
Subspace Discriminant	81.2	30.6	52.9	78.2	86.5
Subspace KNN	84.7	85.3	82.4	70.0	85.9
RUSBoosted Trees	85.3	60.0	53.5	57.1	27.6

Table 8. Comparing Classification Metrics Across All Features [23]

Feature/ MLC Prediction	Resolution	Buffer Length	Bitrate	Dropped Frames	MOS
Fine Tree	98.8	Not ideal	97.6	Not ideal	Not ideal
Medium Tree	98.8	Not ideal	97.6	Not ideal	Not ideal
Fine KNN	Not ideal	91.2	Not ideal	91.8	Not ideal
Weighted KNN	Not ideal	91.2	Not ideal	91.8	Not ideal
Bagged Tree	Not ideal	Not ideal	Not ideal	Not ideal	99.9

Table 9. Resulted Machine Learning Prediction Methods on noisy network data training.

 doi:10.20944/preprints202207.0035.v1

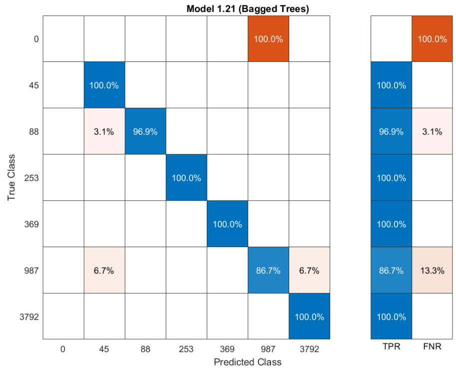

Figure 11. Framework Bitrate Prediction Results

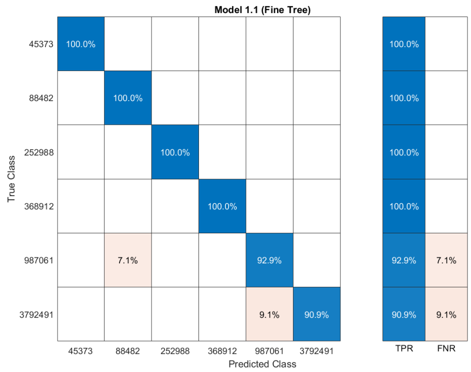

Figure 12. Framework Resolution Prediction Results

11 shows the framework bitrate precision result after choosing bagged trees whereas Figure 12 shows the framework's resolution prediction results after choosing fine tree. The main takeaway from these two figures is to show the differences between the two methods on the same data. This helps our research by identifying and reading correctly predicted true classes and how the methods react to noisy data such as ours.

6. Conclusions

Understanding user-affective network data has become as crucial as the QoE on individual user devices. This paper discusses different features of network level experimentation and the prediction of QoE in multimedia applications. We also discussed how perceivable QoE is linked to resource allocation and traffic engineering at the network level and how emerging programmable networks such as SDN can be used as a tool to improve user feedback and how that data can be used to predict human feedback. An automated data collection and prediction framework is also proposed to harness the capabilities of new network designs and growing availability of computing resources in future networks for fairness-aware content distribution. As shown in Table 8 the ML classification methods are used and compared for the features that are used for prediction and to-be predicted. We conclude that our generated dataset and experimentation framework can support the development of high accuracy machine learning model for QoE estimation. The use of existing neural networks also proved effective with our data but consumed more computational power and time as mentioned in the previous section. Our future work will look into implementations of this QoE/MOS Predictor in a live smart home environment.

References

1. Basil, A.O.; Mu, M.; Al-Sherbaz, A. A Software Defined Network Based Research on Fairness in Multimedia. Proceedings of the 1st International Workshop on Fairness, Accountability, and Transparency in MultiMedia; ACM: New York, NY, USA, 2019; FAT/MM '19, pp. 11–18. doi:10.1145/3347447.3356750.
2. Ke, C.H. P4: Switch: MAC Layer and IP Layer Forwarding.
3. Alsaeedi, M.; Mohamad, M.M.; Al-Roubaiey, A.A. Toward Adaptive and Scalable OpenFlow-SDN Flow Control: A Survey. *IEEE Access* **2019**, *7*, 107346–107379. doi:10.1109/ACCESS.2019.2932422.
4. Basil, A.O. Experimenation datasets and algorithms. https://github.com/AhmedOsamaBasil/PhD, 2022, [https://github.com/AhmedOsa doi:https://github.com/AhmedOsamaBasil/PhD.
5. Moorthy, A.K.; Choi, L.K.; Bovik, A.C.; de Veciana, G. Video Quality Assessment on Mobile Devices: Subjective, Behavioral and Objective Studies. *IEEE Journal of Selected Topics in Signal Processing* **2012**, *6*, 652–671. doi:10.1109/JSTSP.2012.2212417.
6. Chen, C.; Choi, L.K.; de Veciana, G.; Caramanis, C.; Heath, R.W.; Bovik, A.C. Modeling the Time—Varying Subjective Quality of HTTP Video Streams With Rate Adaptations. *IEEE Transactions on Image Processing* **2014**, *23*, 2206–2221. doi:10.1109/TIP.2014.2312613.
7. Ghadiyaram, D.; Bovik, A.C.; Yeganeh, H.; Kordasiewicz, R.; Gallant, M. Study of the effects of stalling events on the quality of experience of mobile streaming videos. 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2014, pp. 989–993. doi:10.1109/GlobalSIP.2014.7032269.
8. Duanmu, Z.; Zeng, K.; Ma, K.; Rehman, A.; Wang, Z. A Quality-of-Experience Index for Streaming Video. *IEEE Journal of Selected Topics in Signal Processing* **2017**, *11*, 154–166. doi:10.1109/JSTSP.2016.2608329.
9. Bampis, C.G.; Li, Z.; Moorthy, A.K.; Katsavounidis, I.; Aaron, A.; Bovik, A.C. Study of Temporal Effects on Subjective Video Quality of Experience. *IEEE Transactions on Image Processing* **2017**, *26*, 5217–5231. doi:10.1109/TIP.2017.2729891.
10. Duanmu, Z.; Rehman, A.; Wang, Z. A Quality-of-Experience Database for Adaptive Video Streaming. *IEEE Transactions on Broadcasting* **2018**, *64*, 474–487. doi:10.1109/TBC.2018.2822870.
11. DASH, I. doi:https://dash.itec.aau.at/dash-dataset/.
12. Yousef, H.; Feuvre, J.L.; Storelli, A. ABR prediction using supervised learning algorithms. 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), 2020, pp. 1–6. doi:10.1109/MMSP48831.2020.9287123.
13. Sani, Y.; Raca, D.; Quinlan, J.J.; Sreenan, C.J. SMASH: A Supervised Machine Learning Approach to Adaptive Video Streaming over HTTP. 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), 2020, pp. 1–6. doi:10.1109/QoMEX48832.2020.9123139.
14. Mao, H.; Netravali, R.; Alizadeh, M. Neural Adaptive Video Streaming with Pensieve. Proceedings of the Conference of the ACM Special Interest Group on Data Communication; Association for Computing Machinery: New York, NY, USA, 2017; SIGCOMM '17, p. 197–210. doi:10.1145/3098822.3098843.
15. Meng, Z.; Chen, J.; Guo, Y.; Sun, C.; Hu, H.; Xu, M. PiTree: Practical Implementation of ABR Algorithms Using Decision Trees. Proceedings of the 27th ACM International Conference on Multimedia; Association for Computing Machinery: New York, NY, USA, 2019; MM '19, p. 2431–2439. doi:10.1145/3343031.3350866.
16. Zhu, Y.; Song, L.; Xie, R.; Zhang, W. SJTU 4K video subjective quality dataset for content adaptive bit rate estimation without encoding. 2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2016, pp. 1–4. doi:10.1109/BMSB.2016.7521936.
17. Ben Letaifa, A. Adaptive QoE monitoring architecture in SDN networks: Video streaming services case. 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017, pp. 1383–1388. doi:10.1109/IWCMC.2017.7986486.
18. Wang, L.; Delaney, D.T. QoE Oriented Cognitive Network Based on Machine Learning and SDN. 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), 2019, pp. 678–681. doi:10.1109/ICCSN.2019.8905343.
19. Abar, T.; Ben Letaifa, A.; El Asmi, S. Machine learning based QoE prediction in SDN networks. 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017, pp. 1395–1400. doi:10.1109/IWCMC.2017.7986488.
20. Abar, T.; Letaifa, A.B.; Asmi, S.E. Real Time Anomaly detection-Based QoE Feature selection and Ensemble Learning for HTTP Video Services. 2019 7th International conference on ICT Accessibility (ICTA), 2019, pp. 1–6. doi:10.1109/ICTA49490.2019.9144867.
21. Al-Issa, A.E.; Bentaleb, A.; Barakabitze, A.A.; Zinner, T.; Ghita, B. Bandwidth Prediction Schemes for Defining Bitrate Levels in SDN-enabled Adaptive Streaming. 2019 15th International Conference on Network and Service Management (CNSM), 2019, pp. 1–7. doi:10.23919/CNSM46954.2019.9012713.
22. INT, I. P.913 : Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment, 2021. doi:https://www.itu.int/rec/T-REC-P.913-202106-P/en.
23. Basil, A.O.; Mu, M.; Al-Sherbaz, A. Novel Quality of Experience Experimentation Framework Through Programmable Network Management. 2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC), 2022, pp. 485–486. doi:10.1109/CCNC49033.2022.9700708.
24. Goto, S.; Shibata, M.; Tsuru, M. Dynamic optimization of multicast active probing path to locate lossy links for OpenFlow networks. 2020 International Conference on Information Networking (ICOIN), 2020, pp. 628–633. doi:10.1109/ICOIN48656.2020.9016438.