Measuring Video QoE from Encrypted Traffic

Giorgos Dimopoulos
UPC BarcelonaTech
gd@ac.upc.edu

Ilias Leontiadis
Telefonica Research
ilias.leontiadis@telefonica.com

Pere Barlet-Ros
UPC BarcelonaTech
pbarlet@ac.upc.edu

Konstantina Papagiannaki
Telefonica Research
dina.papagiannaki@telefonica.com

ABSTRACT

Tracking and maintaining satisfactory QoE for video streaming services is becoming a greater challenge for mobile network operators than ever before. Downloading and watching video content on mobile devices is currently a growing trend among users, that is causing a demand for higher bandwidth and better provisioning throughout the network infrastructure. At the same time, popular demand for privacy has led many online streaming services to adopt end-to-end encryption, leaving providers with only a handful of indicators for identifying QoE issues.

In order to address these challenges, we propose a novel methodology for detecting video streaming QoE issues from encrypted traffic. We develop predictive models for detecting different levels of QoE degradation that is caused by three key influence factors, i.e. stalling, the average video quality and the quality variations. The models are then evaluated on the production network of a large scale mobile operator, where we show that despite encryption our methodology is able to accurately detect QoE problems with 72%-92% accuracy, while even higher performance is achieved when dealing with cleartext traffic.

1. INTRODUCTION

Mobile video will increase 11-fold by 2020, *accounting for 75% percent of total mobile data traffic* [1]. Such rapid growth asserts significant pressure to mobile operators who have to radically rethink and optimize their network.

IMC '16, November 14–16, Santa Monica, CA, USA.
 ISBN 978-1-4503-4526-2/16/11…$15.00
DOI: http://dx.doi.org/10.1145/2987443.2987459

To perform such optimizations and capacity planning, operators have to deeply understand and monitor the offered Quality of Experience (QoE) on video delivery.

Currently, most operators have made significant efforts to facilitate the delivery of media-rich content using techniques such as caching, transcoding, compression and radio resource allocation across users.

At the same time, a significant number of major Internet services have begun to encrypt their traffic. Currently more than 60% of mobile traffic is encrypted, a number that is rapidly rising [2]. Popular video providers such as YouTube, Netflix and Hulu now encrypt a large part of their video content and the trend indicates that most of video traffic will be encrypted soon [3].

While encryption of video content ensures the users' privacy, it significantly impacts the ability of operators to monitor or optimize their network [4]. Practically, with encrypted traffic network operators cannot fulfill essential tasks such as to inspect, protect, prioritize, optimize, compress or balance traffic effectively.

In this paper we present a framework that is able to extract key QoE metrics such as i) stall detection, ii) average representation (resolution), and iii) representation fluctuations in encrypted traffic. More specifically, our contributions are the following:

- We analyze more than 390,000 unique non-encrypted video sessions collected by a web proxy that is deployed on the cellular network of a large provider with more than 10M customers in order to extract insights about video delivery mechanisms and QoE issues.
- We use the insights and the ground truth from the non-encrypted traffic to build a unified QoE measurement method for both adaptive and traditional video streaming over HTTP.
- We then validate our work on encrypted traffic collected from the same network. First we compare the similarities and the differences to the non-encrypted traffic delivery. Furthermore, we setup controlled experiments to verify the accuracy of

the developed model. We demonstrate that the models we developed can identify quality issues from unencrypted data with accuracies between 78% and 93.5% and from encrypted traffic with accuracies between 76% and 91.8%.

- We provide important insights about the information that can be extracted from encrypted traffic. Our results indicate that i) passive measurements from a single vantage point are enough to accurately detect the key factors that affect the users' experience ii) we discuss on the features that are the most significant for detecting each particular problem iii) we demonstrate that client instrumentation is not required.

2. BACKGROUND AND MOTIVATION

2.1 Video Streaming Background

For many content providers HTTP has become the preferred protocol for video delivery over the last few years. HTTP streaming combines advantages such as firewall pass-through and easy network address translation, but also the benefits of TCP, i.e. congestion control mechanisms and reliable packet delivery.

Traditional HTTP Video Streaming

In traditional HTTP video streaming, the video is downloaded as a single continuous file which represents a single quality setting. Moreover, video buffering is employed as an additional measure to compensate for jitter and short-term bandwidth variations.

Typically, each video session can be divided into two buffering phases, i.e. the start-up phase and the steady state [5]. During the start-up phase the player will download the first part of the video as fast as possible to quickly fill the buffer and minimize the initial delay before the playback begins.

Once the buffer has been filled up to a specific threshold and the playback has started, the video session goes into the steady state. This phase is characterized by *ON-OFF* cycles, also referred to as pacing, where the download is paused as soon as the buffer has been filled and resumes when it is reaching depletion.

HTTP Adaptive Streaming (HAS)

In contrast to traditional streaming, HAS videos are split on the server in multiple segments, each one corresponding to a few seconds of playback time. Each segment is encoded in a range of different quality profiles which are defined by the content provider.

Instead of requesting the entire video, the player performs HTTP requests to fetch consecutive segments. The quality profile of the next segment is determined as a function of the throughput with which the previous segment was downloaded and the available seconds of playback in the buffer. In this way, the representation of the video can change dynamically to adapt to changes in the network and minimize stalls.

2.2 Factors that Affect Video QoE

Initial Delay

The initial delay refers to the time spent from the moment the user requests the video until the playback begins. This delay has two components, the network delay and the initial buffering delay. The former can be attributed to factors such as network latency, longer server response times, DNS lookups and/or CDN redirections. The later is caused by the time required to perform the initial fill of the buffer with sufficient video data to allow a smooth playback.

Both Mok et al. [6] and Etoh et al. [7] agree that this factor has the lowest impact on the QoE as users tend to be more tolerant to longer initial delays than other impairments such as stalls or quality changes.

Stalls

Whenever the network throughput is not sufficient for the content to be downloaded faster than the rate that it is consumed, the buffer is depleted and the playback is forced to pause until more data are downloaded and the buffer is filled again.

Hoßfeld et al. [8] showed that not only the frequency but also the duration of the playback stalls which occur due to buffer outages, have a high correlation with poor QoE. Specifically, the authors conclude that a video with 2 stalls of 3 seconds duration each, will lead to significantly lower Mean Opinion Score (MOS).

Moreover, Mok et al. [9] found that the rebuffering frequency has the highest impact on QoE and that a medium rebuffering frequency can result in a MOS lower by 2 points.

In this work, we measure the stalls using the *Rebuffering Ratio* which is expressed as the time spent stalling over the total duration of the video session.

Average Representation Quality

The average quality can be applied only to HAS video sessions, since only in these cases quality representation changes may occur. It is calculated as the average of all the individual qualities of the segments which belong to a video session.

Multiple related works have shown a high correlation between the video representation quality and the user's QoE. In one of these studies [10], the subjective experiments performed in mobile networks have shown that video streams with higher quality representations are linked to better overall QoE.

Representation Quality Variation

Another factor that affects the QoE of adaptive video streaming, is the changes in quality variation. The variation in this case has two dimensions, the frequency of the changes and their amplitude. The frequency is the absolute number of changes that occurred in a video ses-

sion, while the amplitude corresponds to the difference in magnitude between two consecutive qualities.

In [11], the authors investigate how the representation switching amplitude and the switching frequency affect the QoE. Their results show that the switching amplitude has a very high impact on the user experience.

2.3 Problem Statement

Adaptive streaming and encryption are nowadays the default technologies used by the majority of the popular content providers. The widespread adoption of these new technologies has given rise to a new set of challenges for identifying video QoE issues and has rendered previous solutions obsolete.

Deep Packet Inspection (DPI) solutions for extracting quality metrics, such as the video resolution and stall characteristics [12], [13], do not work anymore with encrypted traffic. Moreover, adaptive quality switching has introduced new factors that affect the user's experience, i.e. quality switching amplitude and frequency. However, these factors were not included in previous models for video QoE.

These changes in video streaming technologies, have caused a high demand, not only by network operators but also the by research community, for updated tools and methods for detecting and quantifying quality issues.

Towards this end, this work aims to provide new methods for assessing the different types of impairments that affect the users' QoE from encrypted traffic.

2.4 Challenges

Although many services have already made the migration towards adaptive streaming, their platforms continue to maintain backward compatibility with traditional static streaming. Therefore, one of the main challenges in this work, is to provide a solution which will be compatible with current but also previous video streaming technologies.

Moreover, with end-to-end encryption enabled, a great part of the metrics that were previously available in the network traffic for detecting QoE issues is now becoming inaccessible. For this reason, one of our challenges is to identify the right metrics from the limited amount of information that is provided by encrypted traffic and build the models to detect quality impairments. In order to accomplish that, we need to reverse engineer the video services and rely on machine learning and time series analysis.

Finally, in order to preserve the user's privacy but at the same to make our solution as generalizable as possible, we focus on developing a methodology that will be capable of detecting problems from network traffic alone and will not depend on the instrumentation of devices or video players and therefore it can be easily deployed by operators.

3. DATASET

The set which is presented in this section is constructed from unencrypted data that contains the ground truth for the QoE impairments of each video session. This information is then used to create the predictive models for identifying each impairment type. We then move to a set of encrypted data to validate the previously constructed models using controlled experiments.

3.1 Weblogs

The data is collected from a web proxy that is deployed on the cellular network of a large European provider. The proxy is capable of registering all unencrypted HTTP traffic including IP-port tuples, URI's, object sizes, transaction times, request time-stamps and more. Moreover, each log is annotated with a set of transport layer performance metrics, i.e. bandwidth-delay product (BDP), bytes-in-flight (BIF), packet loss, packet retransmissions and RTT. The BDP is equal to the link's capacity divided by its round-trip delay and represents the maximum amount of bytes that can be transferred by the link at any given time.

The dataset is created from YouTube traffic weblogs which are collected over a period of 45 days spanning from February to April 2016. From all the HTTP traffic that is generated by the service, we keep the weblogs that correspond to video and audio segment downloads and the signalling exchanged between the video player and the service during playback.

All the data is anonymized before the extraction by removing all private information such as user agents, subscriber and handset identifiers, MAC and IP addresses and so on. The only identifier which is preserved is the unique 16-character video session ID which is generated by YouTube. This parameter is described in more detail in Section 3.2.

We find that YouTube is the most suitable candidate among the currently popular video streaming services for developing and evaluating our methodology. The main reasons for this are i) the service's huge popularity which allows the generation of a very rich dataset in a short time window, ii) the diversity of the provided content in terms of video formats, qualities and durations, iii) its popularity among mobile users and iv) the adoption of modern technologies i.e. Dynamic Adaptive Streaming over HTTP (DASH), HTML-based video playback and pacing.

Moreover, most of the popular video sharing services are currently following YouTube's streaming paradigm, adopting adaptive streaming, a variety of supported codecs and HTML-based players.

Note that although Google has in the recent years deployed HTTPS for all of its services including YouTube, we can still observe significant amount of video sessions in cleartext HTTP in our dataset. This is attributed to the fact that many users use legacy devices or players

that either do not support TLS encryption or do not have it enabled by default.

Nevertheless, we verified through experiments in the lab that apart from the encryption which is enabled by default, the delivery mechanism and overall behaviour of the app remains the same with newer devices with modern browsers and the latest version of the app.

In the weblogs, each segment download is generated from the client with a separate HTTP request and therefore we obtain a new entry for each new video chunk. From the list of metrics mentioned above, we also compute the *chunk size* and the *chunk time* that indicates the time when a video chunk arrives at the client, since in our experiments we found they bring relevant information to model the QoE impairments. The complete list of the metrics extracted from the traffic can be found in Table 1.

The final set consists of approximately 390,000 unique video sessions. However, only 3% of these are adaptive streaming sessions. This imbalance is expected since we are able to observe traffic from mainly legacy devices and video players which do not support the more recently adopted adaptive technology.

For the methodology of the stall detection we take the entire dataset, while for the development of the average representation and the representation quality switch detection we only keep the videos that made use of adaptive streaming.

Network Features	Ground Truth (URI)
minimum RTT	chunk resolution
average RTT	stall count
maximum RTT	stall duration
Bandwidth-delay product	video session ID
average bytes-in-flight	
maximum bytes-in-flight	
% packet loss	
% packet retransmissions	
chunk size	
chunk time	

Table 1: Metrics that we extract from the operator's web logs (left column) and the ones that are reverse engineered from the request URIs (right column). The features (left) are available for encrypted and non-encrypted flows whereas the ground truth is only available for non-encrypted sessions.

3.2 Ground Truth

From the meta-data that are passed as parameters in the URIs of the HTTP requests we are able to collect the ground truth that will be used in the evaluation phase. In more detail, these parameters carry three main types of statistics, i.e. generic device and player stats, content stats and playback stats [13].

The generic stats include information about the user's device such as OS, locale, screen resolution, player type and so on. One of the most important parameters here, is the unique video session ID. This ID is a 16-character hash that is randomly generated and it is unique to each session. We use it to identify and group together all the weblogs that belong to the same video session.

The content stats are extracted from the HTTP requests for downloading the individual video segments. One of the the parameters in this group is the 'content type', which indicates if the segment contains video or audio content and the multimedia container that was used to encode it, e.g. MP4, FLV or WebM. 'Itag' is another parameter which is used to specify the bit-rate, frame-rate and resolution of the segment, which we use to obtain the ground truth for the changes in representation quality throughout the session.

Finally, the playback statistics are included in the statistical reports that are periodically sent from the player to Google servers during the playback. Each report contains information that summarizes the progress of the playback since the previous report was generated. Different flags are used in the reports to specify if the video has successfully loaded, if the playback has started, paused or stopped and if there was a stall and how long it lasted. These indicators allow us to discover if a video was played throughout or abandoned and more important, identify the frequency and duration of stalls.

Out of the information that is available in the unencrypted data, we only use the chunk resolution, the stall count and duration and the video session ID (Table 1).

These features will be used as the ground truth for training the detection models in Section 4. After the completion of training phase, the access to the ground truth from unencrypted traffic will no longer be required and even if YouTube removes this information or deploys encryption for all sessions, the methodology will still be applicable.

3.3 Data Preparation

Before starting the analysis, we ensure that any logs that correspond to cached and/or compressed content by the proxy are removed from the dataset.

Next, after the ground truth for the stalls and representation switches is extracted, all the logs that belong to the same video session are identified by the common session ID and are then grouped together.

Thus, each entry in the dataset corresponds to a unique video session which includes information about the total number of stalls and their duration, as well as the characteristics of each chunk such as the quality representation, size, download time-stamp, but also the transport layer statistics like RTT, loss, re-transmissions, BDP and bytes-in-flight for each chunk download.

4. BUILDING THE DETECTION FRAMEWORK

Our approach involves first the development and testing of the detection framework with unencrypted data. As soon as we verify that the constructed models can leverage the cleartext dataset, we can proceed to test the framework with data from encrypted video streams.

As mentioned in Section 2, there are three main types of impairments which may cause the degradation of poor video QoE, the frequency and duration of stalls, the session's Average Representation Quality and the Representation Quality Variation [10].

The initial delay is not considered as part of our video QoE model given its small contribution on the overall user experience as explained in 2.1.

In this section we describe the process of identifying from the limited number of metrics that are offered by the encrypted traffic, those that are the most significant for creating predictive models to detect each of the three types of impairments. An important part of this process is the feature construction that allows the generation of new more powerful features from the already existing ones.

Next, we show that there is a different set of metrics that better describes each type of impairment and contributes more information to the detection model.

In order to generate predictive models for detecting the level of stalling and the average representation, we use Machine Learning (ML) and in particular the Random Forest algorithm and 10-fold cross-validation.

Classification is preferred over regression given that we divide the data in discrete classes in both scenarios and the models are required to identify in which class each video session belongs based on the amount of stalling or the level of the average representation.

4.1 Stall Detection

Feature Construction

From the traffic features described in Section 3 (Table 1), we generate summary statistics, i.e. max, min, mean, standard deviation, 25th, 50th and 75th percentiles for each of the metrics, resulting in 70 new metrics.

Among all the performance metrics that we take into consideration, the chunk size is one of the most important for detecting stalls. If we take an example of a video session were stalling has occurred (Figure 1), we can see the significant changes in the chunk size when the two events take place at the third and the seventeenth second of the video session.

More specifically, whenever there is an outage on the player's buffer that results in a stall, the player will request small chunks which can be downloaded much faster so that the buffer will be filled as soon as possible and the video playback can resume. Then the size of the chunks will gradually increase and remain at a maximum value during the steady state as long as no further issues occur.

Therefore, we understand that we can significantly improve the accuracy of the stall detection model by including the sizes of the chunks in our feature set.

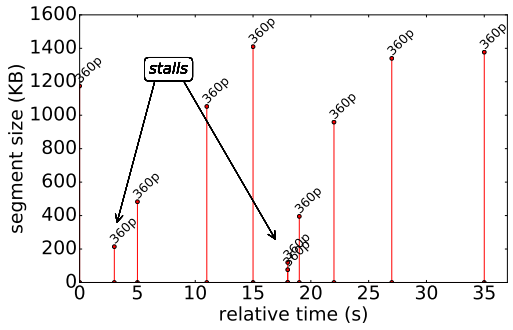

Figure 1: Changes in chunk sizes in a video session with stalls.

After all the required features have been generated, the dataset is then split into sessions without stalls and sessions where at least one stall has occurred. The information regarding the number of stalls observed during a video session and their duration, is the ground truth which is extracted from the meta-data of URIs as mentioned in Section 3.

Figure 2 (left) illustrates the distribution of the number of stalls that occurred per video session. We observe that 12% of all the sessions have suffered from rebuffering events, while about 8% was affected by more than 1 event.

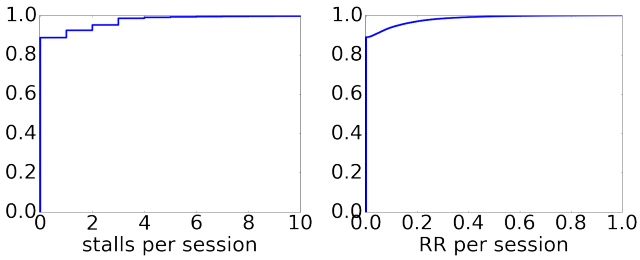

Figure 2: ECDF of number of stalls (left) and rebuffering ratio (right) per session

Labelling

Next, we use the information from the ground truth to label the data and create a predictive model. To do this, first we calculate the re-buffering ratio (RR) for each video session as the ratio of the sum of the duration t_{stall_k} of each of the total K stalls over the duration of the entire session t_{total} (eq. 1)

$$RR = \frac{\sum_{k=1}^{K} t_{stall\_k}}{t_{total}} \quad (1)$$

The sessions are then labelled according to the rule below. The definition of three levels of stalling, i.e. no

stalling, mild and severe, allows a more detailed view of the degree to which the stalls affect the user.

$$Stall\ labels: \begin{cases} \text{“}no\ stalling\text{”}: & RR = 0 \\ \text{“}mild\ stalling\text{”}: & 0 > RR \geq 0.1 \\ \text{“}severe\ stalling\text{”}: & RR > 0.1 \end{cases}$$

The RR threshold for distinguishing mild and severe stalling is set to 0.1, since in their work [14] Krishnan et al. have shown that when the RR is over 0.1, the severity of the stalling causes such a quality degradation that leads the users to abandon the video.

Figure 2 (right) shows the distribution of the RR per video session. We can observe that the sessions with RR equal or greater thatn 0.1 correspond to approximately 10% of the distribution.

Feature Selection

We then proceed to apply Feature Selection (FS) using the Correlation-based Feature Subset Selection (CfsSubsetEval) with the Best First search algorithm to reduce the number of features from 70 to the following four, BDP mean, packet re-transmissions max, chunk size min and the chunk size standard deviation.

The output of the feature selection algorithm reveals that there are three important factors that are correlated with stalling, BDP which is equivalent to throughput, number of retransmissions and chunk size. The limited throughput and increased number of retransmitted packets are QoS metrics which are performance indicators of congested networks and/or networks with limited bandwidth where stalling is more likely to occur.

Table 2 shows the gain of each of the features that were obtained after FS was applied and their respective information gains. The information gain represents the contribution of each feature in the construction of the predictive model. Features with higher information gain have a higher correlation with the problems that we want the model to detect and are used more frequently by the classifier.

The higher gains for the minimum and standard deviation of the chunk size indicate that both these features carry important information for detecting if a video suffered from stalls or not. Smaller chunk sizes correspond to lower quality streams that are frequently selected by the user or the adaptive algorithm in the presence of poor network conditions and limited bandwidth.

On the other hand, larger deviation of the size of chunks is related to sudden changes in the network's performance that in turn lead to quality switches during playback. In both cases the videos which are streamed under these conditions are more prone to stalling due to buffer outages.

The BDP and number of packet retransmissions have a more clear and direct correlation to low bandwidth and congestion scenarios where the speed at which the video buffer is filled is limitted and therefore there is a much higher probability of stalling. These metrics can be beneficial specially for cases of traditional streaming where the video is downloaded over a single connection.

info. gain	feature
0.45	chunk size minimum
0.25	chunk size std. deviation
0.18	BDP mean
0.12	packet retransmissions max

Table 2: Features and respective gains for the stall detection model.

Training and Testing the Predictive Model

In order to avoid biasing the results during the test phase, we balance the number of instances among the three classes before training the classifier. The instances in the classes are then restored to their original numbers for testing.

Overall, the classifier is able to make predictions with 93.5% accuracy. The proposed stall detection model is a significant improvement over previous approaches [15] where the achieved accuracy was approximately 84% for a binary classification. In contrast, our model not only achieves much higher accuracy but it also can predict the severity of the stalling that affected the user.

The output of the test phase of the model in terms of True Positives (TP), False Positives (FP), Precision and Recall can be found in Table 3, while the corresponding confusion matrix is shown in Table 4.

Precision is calculated as the ratio of TP over TP and FP and corresponds to the accuracy with which a certain problem is predicted. Recall is equal to the ratio of TP divided by the total instances in this class and measures the models's ability to correctly identify the QoE issue of a video session from the data set.

From the confusion matrix we can see that the classification errors occur between instances without stalls and those with mild stalls but also between mild and severe. However, significantly fewer misclassifications happen between the severe and “no stall” classes.

Therefore, it is straightforward that the errors occur due to the classifier's inability to correctly identify marginal cases where the RR is close to the RR thresholds we defined for labelling the instances. Hence, instances with RR slightly over zero can be falsely predicted as healthy sessions without stalls and thus increasing the number of FP. The same applies for cases where the RR is marginally over 10%, which can be identified as mildly problematic and vice versa.

In more detail, although some marginal instances belong to different classes, they often have similar characteristics, such as throughput delay and loss. The similarity between instances of different classes can cause confusion to the classifier resulting to the generation of FP.

From Table 3, we can see that the healthy sessions are predicted with higher Precision and Recall when com-

pared to the other two classes. Moreover, the confusion matrix in Table 4 indicates that very few sessions have been misclassified as mildly or severly problematic.

These indicators show that healthy video sessions are streamed in significantly better network conditions as opposed to the problematic ones. This is translated to higher BDP and close to zero packet retransmissions for the vast majority of the instances. Additionally, healthy conditions allow higher quality streams with fewer or no quality switches. The combination of these characteristics allow the algorithm to easily distinguish healthy videos from problematic ones.

The separation of problematic sessions can be more challenging however, which can be verified from respective values in the confusion matrix. Here, in contrast to the healthy cases, there is a much higher number of misclassifications between the videos with mild stalls and those with severe stalls. In these cases, the chunk size often is not sufficient to indicate the amount of stalling. The reason for this is that frequently the minimum video quality is already selected due to limited bandwidth and therefore the minimum chunk size or its standard deviation will not contribute significant information for detecting the amount of stalling that took place during a video session.

Class	TP Rate	FP Rate	Precision	Recall
no stalls	0.977	0.111	0.965	0.977
mild stalls	0.809	0.035	0.816	0.809
severe stalls	0.793	0.009	0.887	0.793
weighted avg.	0.935	0.09	0.934	0.935

Table 3: Classifier's output for the stall detection model

original label	predicted label		
	no stalls	mild stalls	severe stalls
no stalls	97.76%	2.06%	0.18%
mild stalls	14.7%	80.9%	4.4%
severe stalls	4.2%	16.5%	79.3%

Table 4: Stall detection confusion matrix

4.2 Average Representation Detection

Feature Construction

In order to detect the average representation of videos with higher accuracy, in addition to the 10 features that are already available in the dataset, we construct five new ones, i.e. the chunk average size, the chunk size delta, the chunk time delta, the average throughput and the throughput cumulative sum. The chunk resolution is only used for the ground truth and labelling of the instances and not for the construction of the predictive model. Hence, we have a total of 14 features from which we extract the following statistics, minimum, mean, maximum, std. deviation and 5th, 10th, 15th, 20th, 25th, 50th, 75th, 80th, 85th, 90th and 95th percentiles. As a result, the total number of features we end up with is equal to 210.

The chunk average size is calculated from the sizes of all the individual chunks in a video. The size of a chunk has a strong correlation with the respective quality of the video segment. The chunk size delta represents the difference in the size of consecutive chunks while the chunk time delta corresponds to the inter-arrival time of video chunks. These parameters are indicators of representation switches which in turn affect the average representation of the session and will be discussed in more detail in Section 4.3.

Figure 3 presents a video session with a representation switch from 144p to 480p. Each point in the plot represents a video chunk, while the labels above the points indicate the segments' resolutions. The x axis corresponds to the video session relative time and the y axis to the size of the video segments. In this example there is a representation switch from 144p to 480p at $t = 22$ of the time line. This is translated to a significant increase for both chunk Δt and chunk $\Delta size$, which indicates that they can be relevant indicatiors of quality switches.

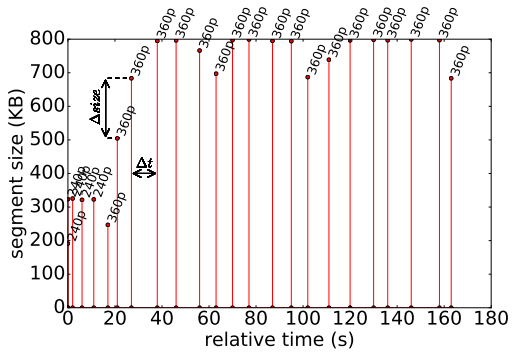

Figure 3: Δt and $\Delta size$ in a video session with a representation switch

The average throughput is calculated from the individual throughputs of all the chunks, while the cusum is their cumulative sum. The later is used as an indicator of variations in throughput.

Labelling

For the detection of the average representation of a video session, it is necessary to categorize the videos in three main categories based on their average resolution, low (LD), standard (SD) and high definition (HD). Given that in our dataset all the observed resolutions take only a few standard values, i.e. 144p, 240p, 360p, 480p, 720p and 1080p, we label all videos with resolutions 144p and 240p as LD, 360p and 480p as SD and all videos with higher resolution as HD.

In the dataset 57% of the videos have LD average quality, 38% have SD quality and only 5% have HD. This is an expected finding in our case where videos

are streamed using limited mobile data plans and on handheld devices that often come whith smaller screens which leads users to opt for LD and SD video qualities.

However, we need to also account for cases where there are representation changes during the playback. For these videos, we calculate the average representation μ from the resolutions of all the segments. We proceed to label the instances in the dataset following the rule below for calculating the Representation Quality RQ.

$$RQ = \begin{cases} HD: & \mu > 480 \\ SD: & 480 \geq \mu \geq 360 \\ LD: & \mu < 360 \end{cases}$$

Feature Selection
The FS is again performed with the aid of CfsSubsetEval and Best First. After the selection there are 15 features remaining out of the initial 210. These features are listed in Table 5, ranked by their respective information gain.

We observe that statistics derived from the chunk size are the ones with the highest rank and represent the vast majority of the 15 features. This is a meaningful and expected result since the chunk sizes are highly correlated with the different representation qualities.

Moreover, the list of features also contains the BDP and the BIF which are proportional to the amount of bytes that can be delivered by the network but also the throughput cusum which is related to the throughput variations throughout the video session.

info. gain	feature
0.41	chunk size 75%
0.39	chunk size 85%
0.38	chunk size 90%
0.37	chunk size 50%
0.33	chunk size max
0.32	chunk avg size mean
0.22	BIF avg max
0.21	cumsum throughput min
0.2	chunk $\Delta size$ max
0.19	chunk size std
0.16	chunk $\Delta size$ std
0.15	chunk Δt 25%
0.06	BDP 90%
0.05	BIF maximum min
0.03	RTT minimum min

Table 5: Features used for the Average Representation detection.

Training and Testing the Predictive Model
The model to predict the average representation quality is again built using ML and Random Forest. The training is done with balanced classes and then the trained model is tested on the entire set. The obtained overall accuracy in this case is 84.5%. The accuracy for each class is provided in Table 6 and the corresponding confusion matrix in Table 7.

Class	TP Rate	FP Rate	Precision	Recall
LD	0.9	0.206	0.845	0.9
SD	0.768	0.106	0.82	0.768
HD	0.756	0.003	0.945	0.756
weighted avg.	0.841	0.156	0.841	0.841

Table 6: Classifier's output for the average representation model

original label	predicted label		
	LD	SD	HD
LD	90%	9.9%	0.1%
SD	22.7%	76.8%	0.5%
HD	6.8%	18.2%	75%

Table 7: Average representation confusion matrix

The accuracies in the later table reveal that our model is able to predict the average quality of LD videos with very high accuracy but with slightly reduced accuracy in the case of SD and HD videos. Nevertheless, the overall but also the individual accuracies remain in high levels, which verify the model's good performance.

When further investigating the accuracy loss however, we identify that its caused by the increased number of misclassifications that occur in the SD and HD classes. More specifically, a considerable amount of SD video sessions is falsely detected as LD, while 18% of HD videos are identified as SD.

This behavior is attributed to the quality downscales that happen during a video session. As a result one part of the video is streamed in higher quality and the part after the downscale is streamed with lower quality. The differences in chunk sizes between the two qualities of a session lead to the incorrect classification of the video. Of course the effects of this phenomenon cannot be observed for LD videos since there is no lower quality to downgrade to and chunk sizes remain consistent throughout the session.

4.3 Representation Quality Switch Detection

Adaptive streaming can adjust the representation of the video during playback in order to compensate for changes in the network conditions and reduce the likelihood of playback buffer outages that lead to stalls. The duration and frequency of the representation changes, also known as Presentation Quality Switch Rate (PQSR), as well as the amplitude of the switch can have a negative impact on the perceived QoE.

Filtering
During the start-up phase, many content providers employ a fast start mechanism that allows them to fill the playout buffer and start the playback as fast as possible, effectively reducing the start-up delay. This short initial part of a video session may have very different char-

acteristics in terms of segment sizes, inter-segment arrival times and throughput when compared to the much longer steady phase.

To reduce the noise introduced by the start-up phase in the detection of resolution variations, we remove the first ten seconds of all video sessions in our dataset. Given that this initial section represents a very small fraction of the entire video session (the average session duration is approximately 180 seconds), we can safely remove it to reduce the noise introduced by the start-up phase while maintaining more than 95% of the session.

Labelling

In order to build a model for quality switching detection, it is necessary to first quantify the switches in terms of frequency and amplitude. To this end, we define two metrics, the time spent in each representation t_r, the frequency of representation switches F and the switch amplitude A.

The switching frequency F is simply calculated as the total number of switches that were observed in a video. The lower the value this metric has, the better the quality of the corresponding video is.

Finally, equation 2 which is based on the work of Yin et al.[16], expresses the switch amplitude A as the normalized sum of all the amplitudes of representation switches between consecutive segments r_k and r_{k+1}. Again, A is analogous to the degradation of QoE since large representation changes which lead to poor QoE will return higher values of A.

$$A = \frac{1}{K-1} \sum_{k=1}^{K-1} |r_{k+1} - r_k| \tag{2}$$

The two metrics are then combined to a single indicator of the representation variation *Var* using linear combination. Next, each instance in the dataset is classified in one of three main categories, no variation, mild variation and high variation, based on the value of *Var*.

Change Detection

During the study of the sessions with many representation changes, we observe that whenever the adaptive algorithm enforces a change in the representation of the video, a new start-up phase is initiated for the new representation. During this phase, the size and inter-arrival times of the segments are reduced significantly until a certain threshold in the playout buffer has been reached and the video download returns to the steady phase.

In the video session in Figure 3, we can see there is a steady state in terms of size and inter-arrival times for the first quality. When the representation switch occurs however, the chunk time delta and size delta are gradually increasing until a steady state is reached again.

Therefore, for the purpose of more accurately capturing the representation changes we use the two features that were used in section 4.2, the segment size delta $\Delta size$ and segment time delta Δt.

The most suitable approach to detect representation changes, is to perform a time-series analysis. This method allows the identification of abrupt changes in the values of different metrics in the dimension of time that are correlated with the switches of representations.

In more detail, our analysis of video sessions with quality switches showed that whenever a change in resolution takes place, a new start-up phase is initiated in order to fill the buffer with data from the new representation as fast as possible. This phase is characterized by video segments with small sizes and small inter-arrival times which will increase gradually until the steady state is reached once again.

We find that the metric which better captures the changes in both the size and the inter-arrival of the video segments, is the product $\Delta size \times \Delta t$. Specifically, the multiplication of the two parameters will combine but at the same time emphasize the effects of each one. Therefore, for each video session in the dataset, we calculate a new time series where each point corresponds to the aforementioned product.

While there are many tools and algorithms for detecting abrupt changes in a time series, we find that the most suitable for the purposes of this work is the Cumulative Sum Control Chart (CUSUM) which was developed by E.S. Page [17].

CUSUM is a change detection monitoring technique which allows the detection of shifts from the mean of a given sample of points in a time series. When a point exceeds an upper or lower threshold then a change is found. In our case, instead of thresholds we use the standard deviation of the output of the change detection algorithm. The standard deviation is capable of capturing the magnitude of the changes that occurred and is an indicator of high variance.

Figure 4, shows the distributions of the standard deviation of the change detection output for sessions with and without variance. We observe that there is significant separation between the two distributions and by defining a threshold at value 500 on the horizontal axis, we are capable of correctly identifying 78% of the sessions without variance and 76% of those that have representation variations.

Apart from the time-series analysis, ML was also considered to develop a model for the detection of representation switches. However, it did not perform as well as the proposed methodology did and for this reason that approach was not considered.

5. EVALUATION WITH ENCRYPTED TRAFFIC

In this section we present and discuss the findings from the evaluation of the models that were developed in Section 4 with encrypted data. This step is important for verifying that the proposed methodology can

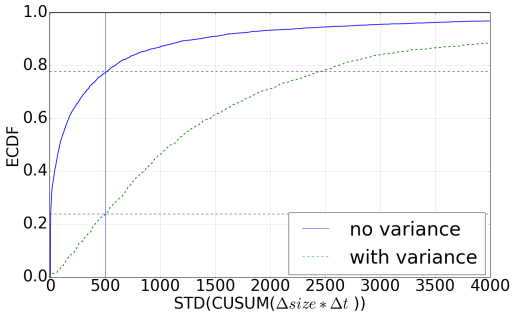

Figure 4: CDF of change detection output for videos with and without resolution changes.

perform with similar accuracy when dealing with encrypted traffic.

5.1 Ground Truth

For the collection of the encrypted traffic, we developed an Android application which is responsible for automatically launching YouTube videos which are randomly selected from the list of the 100 most popular videos on the website [18]. All videos are played using the latest version of the stock YouTube app for Android, where encryption is enabled by default.

Apart from handling the playback of videos, the app has also the capability to extract performance measurements related to the video that is being played. In more detail, by accessing the device's log, it can identify and log the playback status of a video, i.e. if the playback has started, paused, stopped or if a stall has occurred. Therefore, we do not only detect if the video was watched throughout its full length or abandoned earlier, but also identify any stalling events and their duration. This information is used as the ground truth for labeling the data and evaluating the accuracy of the stall detection model.

In order to capture the ground truth related to the representation quality switches we need access to the metadata in the HTTP requests that are responsible for the download of the individual video chunks. However, these requests are encrypted by default by the YouTube application and the required information cannot be captured by means of traffic monitoring.

Although solutions such as Man-in-the-middle (MITM) proxies are common in such use cases for decrypting the traffic generated by the device, we believe that they are not practical since they alter the path between the client and the server, but also change the encryption scheme by establishing two separate TLS connections instead of one.

To make sure that the ground truth for the quality switches is obtained without tampering with the encryption scheme or the traffic between the player and the content server, we reverse engineer the YouTube application and pinpoint the method which is responsible for constructing and performing HTTP requests. Our application then 'hooks' each invocation of this method and extracts its result, which in this case is the full URL of the HTTP request. The URL is then parsed to extract the required ground truth.

Finally, our app will periodically aggregate and send the collected information from the videos to a remote server. The local copy of this information is then deleted from the device to free up space.

5.2 Dataset

Next, the app was installed on a Samsung Galaxy S2 device with a SIM card with an unlimited 3G data plan. The instrumented phone was given to a user who was instructed to carry it at all times for a period of 25 days. The user was motivated to launch the application when moving to increase the probability of QoE issues.

As a result, we generated a dataset for the ground truth and a dataset from the encrypted traffic corresponding to 722 video sessions. Each entry in the ground truth dataset corresponds to a unique segment and the video session ID which the segment belongs to, the timestamp that marks the beginning of the chunk download, a field to indicate if it is an audio or video segment, the total number and duration of the stalls observed in the session and finally its quality representation.

The encrypted traffic data is collected again from the proxy in the form of weblogs. However, since the flows are encrypted, information such as the session ID, the stall characteristics and the quality level of each chunk are not available. Therefore, we only extract the timestamp of the HTTP request, the server IP address and port, the size of the requested object and the TCP statistics which were described in detail in Section 3.1.

Although the session ID is available in the ground truth dataset and it is used to group the video segment statistics in unique sessions, this parameter is missing from the encrypted data. Even so, we find that it is possible to identify the encrypted segments that belong to the same session and group them together.

To achieve this we go through the following steps:

- Identify the traffic that corresponds to a single subscriber and remove all requests that do not belong to YouTube by filtering out those that have domain names not related to the service.

- Next, we look for the unique HTTP traffic patterns that take place at the beginning of a new video session but also after the completion of the playback. These include requests to m.youtube.com and i.ytimg.com which are responsible for downloading multiple web objects such as HTML, scripts and images to construct the video's web page.

- Longer periods without traffic that correspond to the time between consecutive sessions are identified in order to clearly define the beginning and ending of each session.

This methodology has high accuracy as it successfully identified the vast majority of the sessions that were launched during the entire period of the measurements. However, it can be limited in scenarios were the same subscriber launches multiple videos in parallel and not sequentially. Although such cases are quite rare, it can be challenging to identify the segments that belong to the same video session.

Then the two datasets can be easily joined by matching the respective timestamps and the chunk count per session. As a result, the final dataset contains the same metrics that were described in the left column of Table 1. Having the exact same set of features in both datasets is necessary to allow the evaluation of the trained models that were created in the previous section with the new data from the encrypted traffic.

5.3 Dataset Comparison

In this section we characterize the two datasets and make a comparison of the key features. This will help verify that the encrypted YouTube service behaves similarly to the unecrypted and the model built for plain traffic works for encrypted traffic as well.

More specifically, in Figure 5 we present the distributions of the segment size (left) for encrypted and clear-text. The right figure shows the comparison between the two distributions for the segment inter-arrival times.

In the case of the segment size, there is a significant overlap between the two distributions. This indicates that there is a common pattern with respect to the downloaded chunk sizes of the videos in both datasets which can be translated to videos streamed with similar qualities. Only 10% of the segments were larger than 1MB which can be found in HD videos, while the majority of the segment sizes are consentrated at or below 500KB which corresponds to SD video quality.

The distributions for the segment inter-arrival times also have very common characteristics. However, 60% of the encrypted chunks have slightly lower values in comparison with the respective unencrypted data. The shorter times between chunks are indicative of lower bandwidth availability that results in faster depletion of the playout buffer and a more frequent request of new segments. This observation is expected since a large part of the encrypted videos was downloaded while the user was commuting where network conditions can significantly deteriorate.

5.4 Stall Detection

Before evaluating the model for detecting stalls, we repeat the feature construction process described in Section 4.1. However, an automated feature selection like the one employed in the previous section is no longer necessary since we already know the important features that are required to make predictions and the rest are safely removed. Next, the trained model from Section 4.1 is directly tested with encrypted traffic.

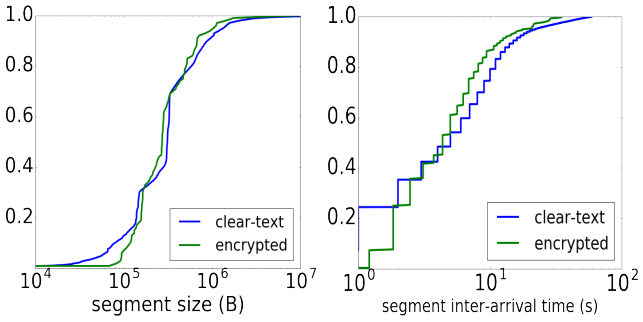

Figure 5: CDF of the segment size (left) and segment inter-arrival time (right) for encrypted and unencrypted traffic.

The resulting accuracy is 91.8% which corresponds to only 1.7% lower performance than the evaluation with unencrypted data. Nevertheless, this is still an excellent result which demonstrates that the training set that we used created a very accurate model that can be applied to encrypted traffic with equal success.

Table 8 shows the evaluation results in terms of Precision and Recall and Table 9 the corresponding confusion matrix. Here we can see that the performance has improved for the videos without stalls, it remained roughly the same for sessions affected by mild stalling but has decreased for the case of videos with severe stalls.

Class	TP Rate	FP Rate	Precision	Recall
no stalls	0.97	0.19	0.96	0.97
mild stalls	0.75	0.04	0.79	0.75
severe stalls	0.64	0.02	0.6	0.54
weighted avg.	0.92	0.16	0.92	0.92

Table 8: Classifier's output for the stall detection evaluation

original label	predicted label		
	no stalls	mild stalls	severe stalls
no stalls	97.2%	2.5%	0.3%
mild stalls	18.6%	75.2%	6.2%
severe stalls	2%	32.4%	65.6%

Table 9: Stall detection confusion matrix

The detection of non-problematic videos is done with higher accuracy than the one observed in Section 4 because there is smaller diversity in the network conditions where the healthy sessions occur. This is attributed to the fact that the majority of these sessions are generated when the user is static either at the office or at home, where the network conditions have a constant performance and as a result, the classifier can more easily identify that these sessions did not have any issues.

The main source of the overall accuracy loss in this evaluation however, is the class of videos with sever stalls. From the confusion matrix it is apparent that

this is a result of the increased number of videos with severe stalls that were falsely detected as mild stalls. This is a problem that was also observed to a lesser extent in the training and evaluation with the unencrypted dataset (Section 4.1).

Although the low performance for the severe stalls class is attributed to the same reasons that were described in the previous section, the further decrease in accuracy originates from the fact that in the new dataset most of the sessions with severe stalls have a Rebuffering Ratio slightly higher than 0.1. Remember that 0.1 is the borderline that was defined to separate sessions with mild and severe stalls. Therefore, it becomes more difficult for the classifier to distinguish to which class these videos belong to.

5.5 Average Representation Detection

The evaluation of the second model for the detection of the average representation is done following the same process as previously. The extended set of features is generated by means of feature construction, followed by the manual removal of the features which do not contribute to the model. This results in the same 15 parameters that were presented in Table 5.

The evaluation is performed with the same approach as previously, where the encrypted dataset is used as a test set for the trained model. The process returns an overall accuracy equal to 81.9% which is approximately 2.5% less than the respective result we got when using the unencrypted dataset in Section 4.2. Again, this is an overall good indicator that the model can perform the detection with almost equally good accuracy when dealing with encrypted traffic.

In Tables 10 and 11, we can see more details regarding the performance of the evaluation per label. Specifically, although the detection of LD and SD videos is done with slightly reduced accuracy, we still get satisfactory performance as we can see from the Precision and Recall values. If we look at the confusion matrix below however, we observe that there is an increase in the LD videos which were misclassified as SD. This is attributed to the fact that in the current dataset the number of 240p videos in the LD category is significantly higher than the 144p. This causes a shift in the distribution of the average quality for this category toward the higher end, which in turn causes the incorrect classification of a percentage of these videos as SD.

Another reason behind the reduction of the accuracy is the reduced detection capabilities for the HD videos. In this case, the Precision and Recall for this class have both reduced significantly. At the same time, from the confusion matrix we see that a significant amount of videos have been incorrectly identified as SD quality. This poor performance is a result of the very small number of videos that are available in the HD class. When combined with the also relatively small number of HD videos that were used to train the model, this results in a class where the training and testing was done with small number of samples and therefore reduced detection capabilities for this class.

This problem can be easily alleviated by introducing a training set that is much richer in HD videos. This will allow the creation of a predictive model which will be based on a more diverse dataset that will be capable of a more accurate detection of the average quality of HD videos with different characteristics.

Class	TP Rate	FP Rate	Precision	Recall
LD	0.845	0.203	0.853	0.845
SD	0.789	0.157	0.775	0.789
HD	0.513	0.003	0.641	0.513
weighted avg.	0.819	0.183	0.819	0.819

Table 10: Accuracies from the evaluation for the average representation detection

original label	predicted label		
	LD	SD	HD
LD	84.5%	15.4%	0.1%
SD	20.4%	78.9%	0.7%
HD	15%	33.75%	51.25%

Table 11: The confusion matrix from the average representation evaluation

5.6 Representation Quality Switch Detection

The last phase of the evaluation is done for detecting quality switches. In this case, there is no trained model that can be directly applied to the encrypted data. In contrast, the methodology relies on the detection of changes that happen in the time intervals between segment downloads and the difference in size between consecutive segments.

In this evaluation there is no requirement for feature construction or feature selection. We only need to calculate the time series of the products $\Delta size \times \Delta t$ for each video in the dataset which is going to be used as input for the change detection algorithm. Next, we apply the change detection on each session and from that we take the standard deviation.

In order to validate the methodology from Section 4.3, we use the same value that was proposed in that section as a threshold for the standard deviation of the change detection output.

$$STD(CUSUM(\Delta size \times \Delta t)) = 500 \qquad (3)$$

According to the proposed methodology, all sessions below the threshold should represent approximately 78% of the sessions without quality switches and the sessions above the threshold should represent 76% of the sessions with quality switches (Figure 4).

Next, the dataset is split into two parts, i.e. the sessions with score below the threshold and those with a

score above it. From the ground truth from the encrypted data, we are able to evaluate if the predefined threshold allows the detection of variance with accuracy equal to the one demonstrated in Section 4.3.

Our analysis reveals that the first part of the dataset consists of 76.9% of videos without any quality change, while in the second part we find 71.7% of the sessions with quality switches. These accuracies are lower by 1.1% and 4.3% respectively as compared to the results from the evaluation with unencrypted data.

The decrease in accuracy for detecting videos with quality switches indicates that the encrypted data consists of videos where the average quality variance is smaller than the one that was observed in the previous section. As a result, the distribution of (3) shifted towards the smaller values and after the threshold was applied, lower percentage of problematic sessions was correctly identified.

6. RELATED WORK

Prometheus [15] uses passive measurements on a mobile network to estimate the QoE of two applications, Video on Demand and VoIP. For the video QoE only Buffering Ratio is considered as a QoE indicator, while the system is evaluated only on unencrypted traffic using binary classification to detect buffering issues with 84% accuracy.

Using similar approaches, OneClick [19] and HostView [20] develop predictive models to detect the QoE of multiple applications including video streaming, using network performance metrics. However, both approaches are limited by the requirement of instrumented devices to capture the feedback from the users.

Hossfeld et al. [11] study the impact of the amplitude and frequency of representation switches on the user experience. The authors re-encoded a video in multiple qualities and introduced different levels and frequencies of switching and performed crowd-sourced experiments to detect correlations with the received MOS from the users. In this work only a single short video was used, which can be considered a very limited representation of the diverse content found in popular services.

In [10] the authors perform subjective tests in mobile networks to assess the impact that the video quality level and quality switching among other factors has on the users' experience. The experiments were conducted with a very limited sample of very short videos, while only the direction of quality switching, i.e. resolution upscaling or downscaling was taken into consideration but not the effects of the amplitude or the frequency.

Finally, the work of Liu et al. [21] investigates three factors that influence the user perceived quality, initial delay, stalling and quality level variation. The authors conducted experiments in the lab with different network conditions in order to derive functions for calculating each of the three impairment factors. The fact that the tests were performed in the lab however, minimizes the generalization of the results to real network conditions and to real streaming services where CDNs and different quality adaptation logics can create different effects in terms of initial delay and quality switches respectively.

Overall, although significant work has been done previously in detecting and quantifying the factors that affect the quality of video streaming, our work is the first that extensively studies these factors in a large scale network using encrypted traffic.

7. LIMITATIONS

The methodology presented in this paper was developed using information from YouTube video sessions that were streamed with the service's current configuration. However, the predictive power of the models responsible for detecting QoE impairments can be limited in the case YouTube changes its video delivery scheme. In such a scenario, the models that were affected by the changes need to be trained and evaluated again with an updated dataset.

Moreover, we do not study the evaluation of the methodology with other video streaming services in order to verify to what extent this approach can be generalized. However, our analysis of other popular video streaming services such as Vevo, Vimeo, Dailymotion and so on, has revealed that they have adopted the same technologies that YouTube is using for content delivery such as adaptive streaming, rate limiting, wide range of codecs and qualities and HTML5-based playback. This common set of characteristics is a strong indicator that our methodology can be generalized to a number of other streaming services and motivates us to include it in the future steps of this work.

8. CONCLUSIONS

In this work we presented a novel framework for detecting from encrypted traffic the 3 key factors that impact both adaptive and classical video streaming QoE, i.e. stalls, average quality and quality switching.

Next, we demonstrated through evaluations on encrypted and unencrypted traffic from a large mobile network, that the proposed models can detect different levels of impairments with accuracies as high as 93.5%.

One of the main findings of the paper is that the changes in size and inter-arrival times of video segments are among the most important indicators of quality impairments. The incorporation of these features in our detection framework resulted in significant improvements in accuracy.

We showed that the framework can perform very well on a real production network using a few key performance metrics from a single vantage point and without the requirement of instrumented clients or additional vantage points, so it can easily be deployed by network operators. The trained models can be then directly applied on the passively monitored traffic and report issues in real time.

9. ACKNOWLEDGMENTS

This work was supported by the Spanish Ministry of Economy and Competitiveness and EU FEDER under grant TEC2014-59583-C2-2-R (SUNSET project) and by the Catalan Government (ref. 2014SGR-1427).

10. REFERENCES

[1] Cisco. "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update". *White Paper*, February 2016.

[2] Sandvine. "Global Internet Phenomena Report". December 2015.

[3] A. Finamore et al. "Is there a case for mobile phone content pre-staging?". In *9th ACM conference on Emerging networking experiments and technologies (CoNEXT)*, pages 321–326. ACM, 2013.

[4] Vasona. "How encryption threatens mobile operators, and what they can do about it". http://goo.gl/fe3xpB. (Accessed on 05/11/2016).

[5] A. Rao et al. "Network Characteristics of Video Streaming Traffic". *7th ACM conference on Emerging networking experiments and technologies (CoNEXT)*, 2011.

[6] R. Mok et al. "Inferring the QoE of HTTP video streaming from user-viewing activities". *1st ACM SIGCOMM workshop on Measurements up the stack (W-MUST)*, 2011.

[7] Z. Guangtao et al. "Cross-Dimensional Perceptual Quality Assessment for Low Bit-Rate Videos". *IEEE Transactions on Multimedia*, 10(7):1316–1324, 2008.

[8] T. Hoßfeld et al. "Quantification of YouTube QoE via crowdsourcing". In *IEEE International Symposium on Multimedia (ISM)*, pages 494–499. IEEE, 2011.

[9] R. Mok et al. "Measuring the quality of experience of HTTP video streaming". In *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pages 485–492. IEEE, 2011.

[10] B. Lewcio et al. "Video quality in next generation mobile networks – perception of time-varying transmission". *IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*, pages 1–6, 2011.

[11] T. Hoßfeld et al. "Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming". In *6th International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 111–116. IEEE, 2014.

[12] R. Schatz et al. "Passive youtube QoE monitoring for ISPs". In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 358–364. IEEE, 2012.

[13] G. Dimopoulos et al. "Analysis of YouTube user experience from passive measurements". In *9th International Conference on Network and Service Management (CNSM)*, pages 260–267. IEEE, 2013.

[14] S. Krishnan et al. "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs". *Networking, IEEE/ACM Transactions on*, 21(6):2001–2014, 2013.

[15] V. Aggarwal et al. "Prometheus: toward quality-of-experience estimation for mobile apps from passive network measurements". In *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, page 18. ACM, 2014.

[16] X. Yin et al. "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP". In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 325–338. ACM, 2015.

[17] ES Page. "Continuous inspection schemes". *Biometrika*, 41(1/2):100–115, 1954.

[18] "YouTube: Most Viewed Videos of All Time". https://www.youtube.com/playlist?list=PLirAqAtl_h2r5g8xGajEwdXd3x1sZh8hC.

[19] K. Chen et al. "OneClick: A framework for measuring network quality of experience". In *INFOCOM 2009, IEEE*, pages 702–710. IEEE, 2009.

[20] D. Joumblatt et al. "Predicting user dissatisfaction with internet application performance at end-hosts". In *INFOCOM*, pages 235–239. IEEE, 2013.

[21] Y. Liu et al. User experience modeling for dash video. In *20th International Packet Video Workshop (PV)*, pages 1–8. IEEE, 2013.

[22] A. Balachandran et al. "Developing a predictive model of quality of experience for internet video". In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 339–350. ACM, 2013.

[23] Z. M. Shafiq et al. "Understanding the impact of network dynamics on mobile video user engagement". In *ACM SIGMETRICS Performance Evaluation Review*, volume 42, pages 367–379. ACM, 2014.