Forecasting YouTube QoE over SATCOM

Matthieu Petrou, David Pradas, Mickaël Royer, Emmanuel Lochin

Forecasting YouTube QoE over SATCOM

Matthieu Petrou
Viveris Technologies
Toulouse, FRANCE
matthieu.petrou@viveris.fr

David Pradas
Viveris Technologies
Toulouse, FRANCE
david.pradas@viveris.fr

Mickaël Royer
ENAC
Toulouse, FRANCE
mickael.royer@enac.fr

Emmanuel Lochin
ENAC
Toulouse, FRANCE
emmanuel.lochin@enac.fr

***Abstract*—We investigate the feasibility of using machine learning methods for predicting the Quality of Experience (QoE) of end users in the context of video streaming over satellite networks. To achieve this, we analyzed QoE and traffic data from 2,400 YouTube video sessions over emulated geosynchronous (GSO) satellite links. The objective is to determine whether existing learning methods, originally developed for wired or mobile networks, can be adapted to accurately predict key QoE factors over SATCOM. We particularly investigate a specific existing framework, which achieves outstanding performance in predicting resolution and initial delay. However, we point out some discrepancies in their hypothesis, leading to optimistic forecasting results. We then refine their methodology to ensure a complete independence between training and test datasets, leading to a fairer QoE video streaming forecast over satellite networks.**

***Index Terms*—SATCOM, QoE, QoS, network monitoring, HTTP adaptive video streaming, machine learning, deep learning**

I. Introduction

Stored streaming videos ('video sessions') represent an important part of global traffic share. The Sandvine report of 2020 [1] reveals that this is also the case over satellite links, with video traffic representing 40% of the total volume. Furthermore, this report shows the importance of YouTube, which is currently the main source of traffic with more than 16% of total traffic over satellite networks. Therefore, assessing the Quality of Experience (QoE) of end users is crucial for network operators.

However, Internet Service Providers (ISP) do not have access to the client-side application and are unable to directly measure the QoE of their customers. Moreover, most of today's traffic is encrypted, preventing the use of Deep-Packet-Inspection (DPI). There are several studies proposing methods to predict streaming video QoE [2]–[4] but to the best of our knowledge, there are no studies for GeoSynchronous Orbit (GSO) satellite links.

GSO satellite networks are a keystone of today's global network. They provide connection to isolated areas and zones that lack effective communication infrastructure. They are also critical solutions in disaster-stricken areas, such as in natural disaster events where terrestrial network systems are damaged. GSO satellite networks have specific characteristics that affect the overall connection, with high latency being the main characteristic due to the altitude of the satellites, which induces a round-trip time (RTT) of around 600 ms. Therefore, existing video QoE predictive models cannot be directly applied to satellite networks.

The first objective of this study is to monitor QoE and network traffic to create a dataset suitable for GSO networks, available in open source to the research community. Additionally, we seek to determine whether the use of previous methodology to build a predicting model can be applied to this dataset. With a dataset of 2,400 video sessions, we cover a wide panel of video lengths (duration) and scenarios. Finally, this study emphasizes the crucial role of ensuring thorough research practices, as we uncover that a prior study reported misleading results.

The remaining part of the paper proceeds as follows: Section II details related research, and Section III deals with the methodology used for this study and presents our collected dataset. Section IV presents and analyzes the results, with a discussion in Section V. Section VI concludes.

II. Background

In this section, we cover the main definition of Quality of Experience (QoE) and how it can be assessed.

QoE is the quality perceived by end users. There are numerous influencing parameters (IFs), related to the user, context, content, and system that all impact the QoE [5]–[7]. From the ISP point of view, they are limited to monitoring and control of system related IFs, and more precisely those of Internet connections. In this context, these IFs are the Quality of Service (QoS), such as jitter, packet loss and throughput. The QoE of video sessions is often evaluated with the Mean Opinion Score (MOS), which is a rating between 1 and 5, with 5 being the best QoE.

The QoE of video sessions is mainly degraded by stalling, initial delay, resolution, and resolution deterioration [8], which are related to the QoS.

'Stalling' is the moment at which the video stops running to refill. Stalling phases are the most impacting events on QoE [8], and as a result, YouTube prevents them from occurring as much as possible: the YouTube algorithm prefers to reduce the resolution or have a longer initial delay than to have stalling phases occur.

'Initial delay' is the phase before the video starts, after being requested by the user. In this work, we use the first packet addressed to YouTube servers as the first request, and also take into account the download of YouTube's webpage. It is

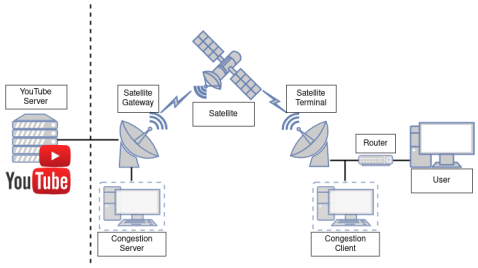


Fig. 1. Topology of our test bench

important to note that increasing the initial delay to reduce or prevent stalling improves the QoE [9].

'Resolution' corresponds to the number of pixels displayed, which is usually the number of vertical pixels that indicate the resolution of the image ($144p$, $240p$, $360p$, $480p$, $720p$, and $1080p$). Resolution variations also impact QoE, and it appears that users prefer a constant lower resolution than a high number of switches between resolutions [8].

Table I summarizes some of the most relevant previous studies related to QoE prediction. We only include the previous studies that are able to predict QoE in real time, as they can be used in concrete applications.

III. Methodology

The section below describes the experimental setup and scenarios used. It also briefly presents how data are processed and analyzes the collected dataset. Finally, it introduces the machine algorithms used in this work. For the sake of the reproducibility of these experiments, our data are available on github[1].

A. Experimental Setup

An emulated satellite link connects the user to the Internet. Satellite emulation is driven by OpenSAND [10], [11], which is an open-source end-to-end satellite communication system emulator. OpenSAND emulates SATCOM systems with a fair degree of realism [12]. Three components constitute the satellite link: a satellite terminal, a GSO satellite, and a satellite gateway. The satellite gateway is directly connected to the Internet and the user is connected behind the satellite terminal through a router. The link between the user's router and the satellite terminal allows us to add packet loss for specific scenarios. Additional clients and servers are added behind the satellite terminal to generate congestion. Fig. 1 depicts this topology. Traffic data are captured on the satellite terminal and the YouTube data are collected from the user.

OpenBACH [13] is an open-source network metrology test bench that orchestrates tests. An OpenBACH script collects YouTube metrics from the SATboost plugin [14], which collects information from YouTube's 'Stats for Nerds' interface. We only use the YouTube collecting feature of SATboost.

[1]https://github.com/viveris/satcom-qoe-dataset

B. Scenarios

We use two forward link capacities on the forward link, $1\,Mb/s$ and $12\,Mb/s$, as in previous studies [15], [16], and because they represent realistic public satellite Internet access. Both capacities have a one-way delay of 250 ms over the emulated satellite link. As OpenSAND cannot reduce the available capacity below 4 Mb/s, to obtain the 1 Mb/s capacity we use the 12 Mb/s configuration and add a prioritized UDP flow of 11 Mb/s, generated by Iperf3 (v3.10.1).

To diversify our data set, we include different scenarios, with additional YouTube users on the client side, with prioritized UDP flows or with different loss conditions on the satellite link.

C. Data post processing

In this subsection, we briefly present the computation of labels and inputs used in the machine learning (ML) algorithms.

1) QoE metrics: using collected YouTube metrics, we compute labels to predict the initial delay and the resolution:

- 'Video Has Started': in order to predict the initial delay, we assess whether or not the video has started. To do so, we consider the viewed frames given by YouTube to compute when the video starts playing; and
- 'Resolution': the resolution played is directly given by YouTube. When a resolution change occurs, the resolution played during the last second cannot be known. Therefore, and as has been done in previous work [4] we remove the resolution label for time slots with resolution changes. For the same reason, we also remove the resolution label when a stall occurs.

2) Network metrics: timestamp, size, and both IP source/destination packets are collected. Using these data, we use the ViCrypt paper [3] approach to compute features. This approach considers what has occurred in three different time windows: the last second, the last 3 s, and the whole session. However, we remove TCP and UDP related features, as they do not provide any improvement to our results. As a result, each time slot has 199 features, with 66 features per time window and one final feature to count the time slots.

D. Data analysis

We perform the test with 40 unique videos, with a time length spread from 30 s to 17 min 32 s. We choose these time lengths to include a heterogeneous representation of what is available to users. Each video is available in the same resolutions: $144p$, $240p$, $360p$, $480p$, $720p$, and $1080p$, and has between 24 and 30 frames per second. This corresponds to a total of 2400 video sessions, for a total of $1,306,761$ time slots of one second that represents more than 15 days.

In the dataset, we consider the initial delay as the sum of the delay in loading the YouTube webpage and the delay in starting the video. Fig. 2 represents a Cumulative Distribution Function (CDF) of collected initial delays. No video starts in less than 10 s, and 50% of monitored videos start within 27 s, which are mainly composed of 12 Mb/s sessions.

TABLE I
OVERVIEW OF PREVIOUS WORK RELATIVE TO QOE PREDICTION

Reference	YouTube Platform	Prediction	Real Time	Video sessions
Requet [2]	YouTube mobile and desktop	Buffer Warning, Buffer Filling Phase, Resolution	Yes	600
ViCrypt [3]	YouTube mobile and desktop	Initial Delay, Stalling, Resolution, Bitrate	Yes	15,000+
Loh 2021 [4]	YouTube mobile	Resolution, Initial Delay, Buffer Filling Phase, Stalling	Yes	13,000+
Our work	YouTube desktop over SATCOM	Resolution, Initial Delay, Stalling	Yes	2,400

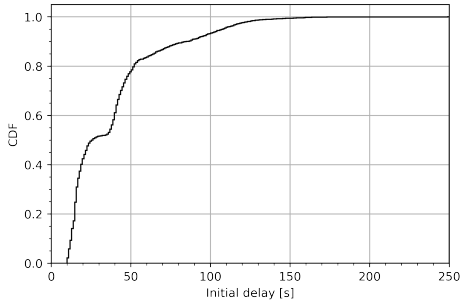


Fig. 2. Distribution of initial delay

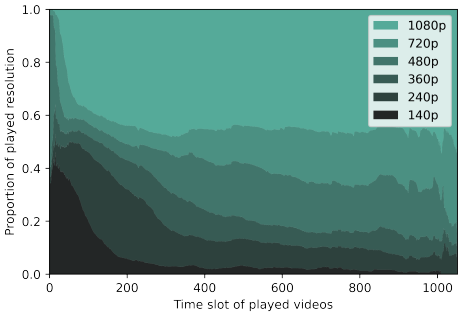


Fig. 3. Distribution of resolutions for each time slot of videos played.

TABLE II
OVERVIEW OF RESOLUTIONS IN DATASET

Resolution category	Resolution	No. of Time Slots	Percentage
Low	144p	111,611	9.39%
	240p	177,710	14.95%
	360p	104,710	8.81%
Medium	480p	170,296	14.33%
	720p	140,080	11.78%
High	1080p	484,320	40.74%

Table II summarizes resolution in the dataset. As the tests are carried out in 'auto' mode, which means that the application determines the downloaded and displayed resolution, we have an unbalanced resolution dataset, with more than 40% of time slots in $1080p$. Resolution is a performance metric, however, a less specific metric could be more relevant to the ISP. Hence, we also consider resolution by category : 'Low' corresponds to resolutions $144p$, $240p$, and $360p$; 'Medium' corresponds to resolutions $480p$ and $720p$; and 'High' corresponds to the resolution $1080p$. Respectively, these resolution categories represent 33.15%, 26.11%, and 40.74%.

Fig. 3 provides the proportion of resolution for each time slot of played videos. The most interesting aspect of this graph is the differences between the resolution distributions over time. We can clearly see that most of the $144p$ resolutions played are within the first 150th time slots. Then, the $1080p$ resolution quickly represents a significant portion of the resolutions played. $360p$ and $720p$ resolutions are under-represented in the first 250th time slots and are more important in later video sessions. Finally, this figure shows that videos start either with a resolution of $144p$ or $480p$.

E. Machine Learning Algorithm

To ensure the accuracy of the results, we take eight unique videos, equally distributed over the different time durations, and use them as a test group. Therefore, we train the machine learning models on 32 different unique videos, and test them on unique videos on which they have never been trained. We use three different machine learning algorithm approaches :

- $LSTM199$: a LSTM (Long Short-Term Memory) using the whole feature set (199 features);
- $LSTM67$: a LSTM using only the feature of the current second (67 features). This approach aims to reduce the computational time of both the features and the algorithm to see if we can achieve comparable result; and
- RF: a random forest of 100 trees and the whole feature set (199 features).

ViCrypt paper [3] uses the Random Forest algorithm with only 10 trees and has good results. We add LSTM as it is an efficient algorithm to predict time dependent data, which could have better results than RF. We denote that, as is conventionally carried out for the LSTM algorithm, we standardize the features for $LSTM67$ and $LSTM199$ (which it is not necessary nor done for the RF).

IV. RESULTS

This section presents the prediction performance for initial delay and resolution.

A. Initial Delay

As discussed in Section III, we consider the initial delay to be a Boolean, as in 'the video has started'. Therefore, with videos longer than the initial delay, almost 93% of the time slots indicate 'Yes''. Table III represents the performance of

TABLE III
VIDEO HAS STARTED PREDICTION RESULTS

		Precision	Recall	F1
LSTM199	No	97.61%	97.65%	97.63%
	Yes	99.82%	99.82%	99.82%
	Average	98.72%	98.73%	98.72%
	Weighted avg	99.66%	99.66%	99.66%
LSTM67	No	97.44%	97.46%	97.45%
	Yes	99.81%	99.80%	99.80%
	Average	98.62%	98.63%	98.63%
	Weighted avg	99.64%	99.64%	99.64%
RF	No	97.82%	95.86%	96.83%
	Yes	99.68%	99.84%	99.76%
	Average	98.75%	97.85%	98.30%
	Weighted avg	99.55%	99.55%	99.55%

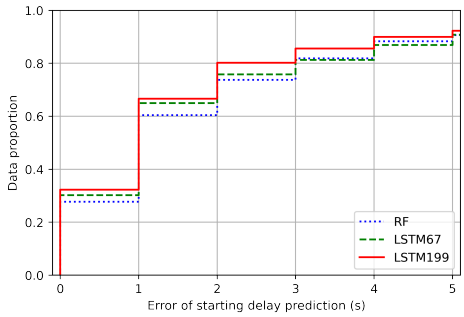


Fig. 4. Distribution of initial delay prediction error: $|predicted - truth|$

ML algorithms for initial delay. An average F1 score of 98% means that we can predict with a great level of exactitude. With more details on accuracy, Fig. 4 represents the CDF of the absolute difference between predictions and ground truths. Results show that there is no important difference between the three algorithms. However, $LSTM199$ is marginally more accurate than the other two with perfect prediction in more than 30% cases. RF is behind both LSTM algorithms, but three models can predict the initial delay in 90% of cases with an error up to $5\,s$.

B. Resolution

Predicting the resolution is more difficult than predicting the initial delay. Performance and confusion matrices of the ML algorithms are presented in Table IV and Fig. 5, respectively. Table IV clearly shows the significant difference between LSTM and RF results. Results obtained also show that the $LSTM199$ F1 score is better than $LSTM67$. It is important to note that with both LSTM algorithms, in more than 90% of the cases, predicted resolutions are within one resolution of actual resolutions.

C. Resolution by category

Table V represents the performance of the resolution category prediction. As we reduce the number of classes with the resolution categories, it eases the prediction of the resolutions and provides a fair indication of the quality obtained by the users. Contrary to the resolution, there is no important difference between the performance of LSTM and RF. In this case, $LSTM67$ has the best average F1 score. Our predictions are acceptable with $LSTM67$, $LSTM199$, and RF with an average F1 score above 90%.

TABLE IV
RESOLUTION PREDICTION RESULTS

		Precision	Recall	F1
LSTM199	144p	86.42%	81.93%	84.11%
	240p	79.40%	79.22%	79.31%
	360p	45.82%	47.28%	46.54%
	480p	48.35%	49.08%	48.71%
	720p	60.43%	67.68%	63.85%
	1080p	98.27%	94.99%	96.60%
	Average	69.78%	70.03%	69.86%
	Weighted avg	79.21%	78.48%	78.80%
LSTM67	144p	82.41%	77.25%	79.75%
	240p	72.96%	74.86%	73.90%
	360p	41.68%	39.85%	40.75%
	480p	47.91%	61.64%	53.92%
	720p	64.06%	54.13%	58.68%
	1080p	97.68%	95.57%	96.61%
	Average	67.78%	67.22%	67.27%
	Weighted avg	77.61%	76.74%	77.00%
RF	144p	78.48%	75.56%	76.99%
	240p	69.59%	69.50%	69.55%
	360p	36.98%	36.95%	36.97%
	480p	39.25%	46.17%	42.43%
	720p	50.99%	37.91%	43.49%
	1080p	93.07%	96.65%	94.83%
	Average	61.39%	60.46%	60.71%
	Weighted avg	71.69%	72.00%	71.67%

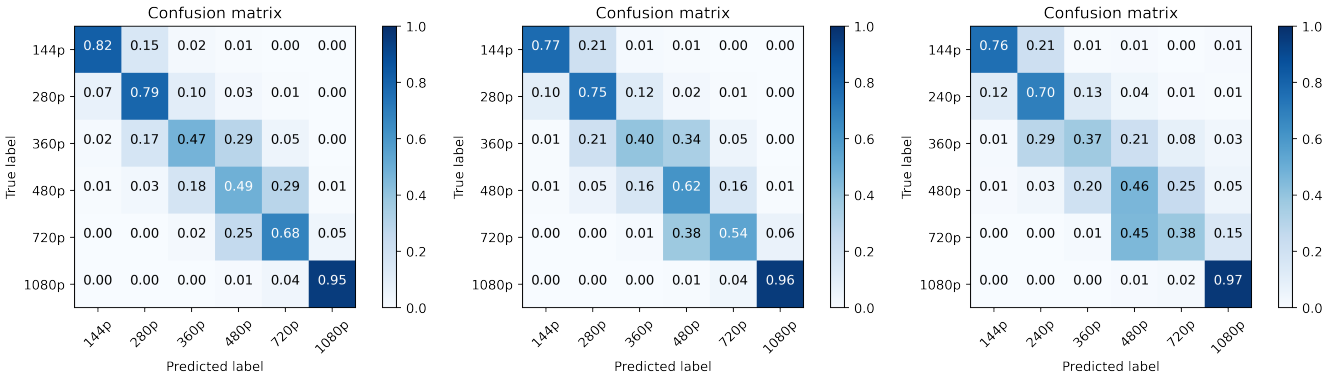
TABLE V
RESOLUTION CATEGORY PREDICTION RESULTS

		Precision	Recall	F1
LSTM199	Low	92.93%	90.32%	91.60%
	Medium	81.34%	85.94%	83.58%
	High	96.70%	95.65%	96.17%
	Average	90.32%	90.64%	90.45%
	Weighted avg	91.60%	91.42%	90.45%
LSTM67	Low	93.11%	92.25%	92.68%
	Medium	84.51%	86.47%	85.48%
	High	97.20%	96.58%	96.89%
	Average	91.61%	91.77%	91.68%
	Weighted avg	92.65%	92.59%	92.62%
RF	Low	91.25%	85.20%	88.12%
	Medium	83.22%	87.40%	85.26%
	High	95.12%	95.45%	95.28%
	Average	89.86%	89.35%	89.56%
	Weighted avg	90.24%	90.12%	90.14%

V. DISCUSSION

In this section, we compare the results with the state of the art.

Regarding the initial delay, we have a prediction close to the work of both Wassermann (ViCrypt) [3] and Loh [4]. ViCrypt has perfect prediction in less than 40% of sessions, while we have this level of accuracy in more than 30% of sessions. Moreover, we have at most 2 seconds of error for 80% of sessions, while ViCrypt has the same in 70% of their sessions. For Loh's work, in more than 80% of sessions, their prediction



(a) $LSTM199$ confusion matrix (b) $LSTM67$ confusion matrix (c) RF confusion matrix

Fig. 5. Confusion matrix of resolution predictions

error is less than 1.5 seconds. In both cases, we are close to their accuracy in predicting the initial delay.

If we compare the F1 score of resolution prediction, it is clear that we score below both the work of Loh and Wassermann and higher than that of Requet [2]. We use a different approach from Loh's work, which may explain the 5% difference, however, we use the ViCrypt approach, which should be closer to their results (rather than 20% below). A possible explanation may be the difference between our methodologies and those used by Wassermann. Wassermann used a five-fold cross-validation to obtain their results, which requires the mixing of all data. As video session data are time-related, this approach can lead to training models on data correlated to the test dataset. Hence, by mixing video session data, Wassermann trains their models using time slots from the past and future of the test dataset. As a matter of fact, the independence between the training and the test datasets is not respected anymore. Therefore, their models would certainly obtain worse results on video sessions that are not mixed with their training dataset. Compared to our approach, we choose to ensure complete independence between training and test datasets by not mixing video session data. To confirm this, using five-fold cross-validation, we obtain an average F1 score of more than 95% for the resolution prediction with the RF.

As none of these previous studies have considered resolution as a category, we cannot compare those results.

VI. Conclusion

In this work, we monitor 2,400 YouTube sessions to estimate QoE metrics from packet traces. To predict QoE metrics, we use the same feature processing as ViCrypt [3]. We apply and compare three ML models to predict the initial delay, the resolution, and the resolution categories. Results show that LSTM algorithms have better or equivalent results compared to RF, and also demonstrate that LSTM marginally gains accuracy by using features from trending and session windows, except for the resolution categories. We highlight the importance of independence between training and test datasets.

We conclude that it is possible to apply approaches used with terrestrial networks to GSO satellite links, with only a few changes, so as to predict the QoE of YouTube end users. We intend to use this method in our future industrial solutions. Our future work will also focus on predicting other QoE metrics, such as stalling and the buffer state. Further interests include performing tests in a Low Earth Orbit (LEO) satellite environment and investigating other potential approaches to predict QoE metrics.

VII. Acknowledgements

This work was supported by Viveris Technologies. The authors wish to thank Victor Perrier for his help.

References

[1] Sandvine. The global internet phenomena report covid-19 spotlight, 2020.
[2] Craig et al. Gutterman. Requet: real-time QoE detection for encrypted YouTube traffic.
[3] Sarah Wassermann, Michael Seufert, Pedro Casas, Li Gang, and Kuang Li. ViCrypt to the rescue: Real-time, machine-learning-driven video-QoE monitoring for encrypted streaming traffic. 17(4).
[4] Frank Loh, Fabian Poignée, Florian Wamser, Ferdinand Leidinger, and Tobias Hoßfeld. Uplink vs. downlink: Machine learning-based quality prediction for HTTP adaptive video streaming.
[5] Kjell et al. Brunnström. *Qualinet White Paper on Definitions of Quality of Experience*. 03 2013.
[6] Parikshit Juluri, Venkatesh Tamarapalli, and Deep Medhi. Measurement of quality of experience of video-on-demand services: A survey. 18(1).
[7] Khadija Bouraqia, Essaid Sabir, Mohamed Sadik, and Latif Ladid. Quality of experience for streaming services: Measurements, challenges and insights.
[8] Michael et al. Seufert. A survey on quality of experience of HTTP adaptive streaming. 17(1).
[9] T. et al. Hossfeld. Initial delay vs. interruptions: Between the devil and the deep blue sea.
[10] Opensand. https://www.opensand.org/.
[11] E. Dubois et al. Opensand, an open source satcom emulator. Kaconf, 2017.
[12] Antoine Auger, Emmanuel Lochin, and Nicolas Kuhn. Making Trustable Satellite Experiments: an Application to a VoIP Scenario. In *89th IEEE Vehicular Technology Conference*, Kuala Lumpur, Malaysia, April 2019. IEEE.
[13] Openbach. https://www.openbach.org/.
[14] Satboost add-on. https://github.com/CNES/satboost/.
[15] Guilloteau Romain, Pradas David, Pelat Guillaume, and Kuhn Nicolas. Recommendations on using VPN over SATCOM. *CoRR*, abs/2111.04586, 2021.
[16] Nicolas Kuhn, Francklin Simo, David Pradas, and Emile Stephan. Evaluating BDP FRAME extension for QUIC. *CoRR*, abs/2112.05450, 2021.