Identifying the Root Cause of Video Streaming Issues on Mobile Devices

Giorgos Dimopoulos*, Ilias Leontiadis†, Pere Barlet-Ros*, Konstantina Papagiannaki†, Peter Steenkiste‡

*UPC BarcelonaTech, Barcelona, {gd, pbarlet}@ac.upc.edu
†Telefonica Research, Barcelona, firstname.lastname@telefonica.com
‡Carnegie Mellon University, prs@cs.cmu.edu

ABSTRACT

Video streaming on mobile devices is prone to a multitude of faults and although well established video Quality of Experience (QoE) metrics such as stall frequency are a good indicator of the problems perceived by the user, they do not provide any insights about the nature of the problem nor where it has occurred. Quantifying the correlation between the aforementioned faults and the users' experience is a challenging task due the large number of variables and the numerous points-of-failure.

To address this problem, we developed a framework for diagnosing the root cause of mobile video QoE issues with the aid of machine learning. Our solution can take advantage of information collected at multiple vantage points between the video server and the mobile device to pinpoint the source of the problem. Moreover, our design works for different video types (e.g., bitrate, duration, ..) and contexts (e.g., wireless technology, encryption, ..) After training the system with a series of simulated faults in the lab, we analyzed the performance of each vantage point separately and when combined, in controlled and real world deployments. In both cases we find that the involved entities can independently detect QoE issues and that only a few vantage points are required to identify a problem's location and nature.

CCS Concepts

•Networks → Network performance analysis; Network measurement;

CoNEXT '15 December 01-04, 2015 , Heidelberg, Germany
 ISBN 978-1-4503-3412-9/15/12…$15.00
DOI: http://dx.doi.org/10.1145/2716281.2836109

1. INTRODUCTION

The use of mobile devices for streaming video is becoming increasingly popular. According to Cisco, mobile video traffic is estimated to increase 14 times between 2013 and 2018, resulting in 69% of the total mobile data [1]. Moreover, YouTube reports that currently mobile makes up for 50% of its global watch time [2].

However, due to the heterogeneity of the devices and networks between the content server and the client, it is often difficult to detect whether the user is experiencing video QoE issues and to identify the root cause of any problem. Apart from typical network-related problems such as delay, congestion or limited bandwidth, video streaming on mobile devices may also suffer from the device's hardware limitations, high load on the endpoints and problems in the wireless medium.

Our goal is to design a system that not only detects the existence of video QoE problems, but also identifies their root cause, e.g., the nature of the problem and where it is located along the data delivery path. To achieve this, we propose to i) place measurement probes at few, key vantage points (VPs) along the path, ii) collect and construct a set of performance features that are agnostic to the noisy and heterogeneous video delivery mechanisms and iii) build a machine learning model to associate these features with poor QoE and perform root cause analysis (RCA).

We initially train and test our system in a controlled environment where we induce common faults. Next, we evaluate whether the trained model is effective in the real world through in-the-wild experiments on YouTube and other video services over the public Internet. Our results indicate that each of the involved parties (users, ISPs, content delivery networks and content providers) can independently identify the existence of poor QoE. Furthermore, only a few vantage points are required in order to pinpoint the exact nature and location of the problem. Finally, the results emphasize the importance of instrumenting end-devices, as metrics collected at the

mobile devices can already identify the vast majority of the faults.

This paper makes the following contributions:

- We designed and implemented a diagnostic tool for video streaming. The system can use a wide variety of network and hardware measurements collected at one or more vantage points along the video path to identify and pinpoint the root cause of detected QoE problems.
- We designed a supervised-machine learning model that uses feature construction and selection to make the diagnostic system both general and practical in real world environments, e.g., where different types of videos and streaming techniques, and wireless technologies are used.
- We combine controlled and in-the-wild measurements to show how much metrics and vantage points contribute to identifying and locating the cause of QoE problems.
- More surprisingly, we show that training our RCA model in the lab is sufficient to lead to more than 80% diagnosis accuracy in the wild, with users accessing video content hosted by us or commercial services like YouTube, and through both cellular and WiFi networks.

The remainder of the paper is organized as follows. Sections 2 and 3 motivate and describe our design. We present the results of controlled and in-the-wild experiments in Sections 4 through 6. Finally, we discuss practical implications, related work and our conclusions.

2. APPROACH AND CHALLENGES

System model - In a typical video streaming session on a mobile device from a popular service such as YouTube, the video data is downloaded from a content server in a Content Distribution Network (CDN). As shown in Figure 1, the video stream is first transferred through Internet Backbone links to the client's ISP. Next, the data is downloaded to the mobile device over a broadband link connected to a home gateway/Access Point or a cell tower depending on the client's connection type.

Each hop in the path may suffer from impairments that can affect the smooth delivery of the video and therefore the user's experience. Congestion or bandwidth bottlenecks in the local or remote network segments, high load on the endpoints and problems in the wireless medium are some of the most significant issues that cumber the performance of video streaming services and contribute in the user's QoE degradation. The goal of the project is to develop a tool that not only identifies the existence of a video QoE problem, but is able to identify its location and its root cause.

Approach - Machine learning has been widely used

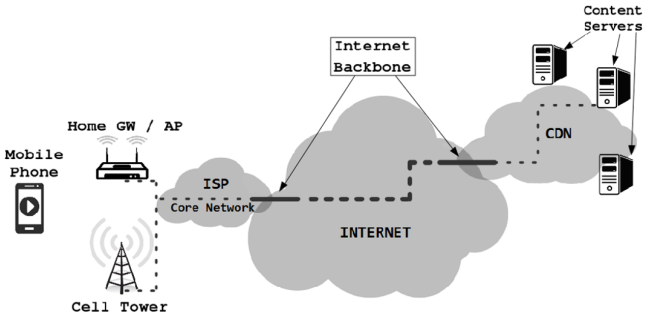

Figure 1: Video data path in the real world.

to help solve complex problems, including fault diagnosis and analyzing the QoE of video, so it is a natural starting point for our work. However, ML alone is not sufficient. For example, based on data collected on the mobile phone, it may be possible to learn that poor QoE is caused by "low bandwidth", but it will not be possible to identify what network is at fault, e.g., the wireless link versus the Internet backbone.

To help both identify and pinpoint failures that may cause QoE issues during playback, we propose to place measurement probes at multiple vantage points (VPs) along the path. Collecting data at multiple points will provide a system-wide view that can help us isolate performance metrics for different segments and devices. In an ideal world, we would be able to collect measurements on every device, but this is not practical so we focus on a scenario with three VPs, the two endpoints and the wireless AP.

Challenges - While there has been some research on diagnosing video QoE, as discussed in Section 8, we are not aware of any work that uses a combination of network and hardware metrics collected across multiple VPs. This novel approach introduces a number of challenges.

First, the amount of data that can be collected across the vantage points is overwhelming. We use feature construction and selection to identify the most useful features (Section 3.2).

Second, providers offer video with different quality, encoding, and duration, and also use a variety of video delivery mechanisms such as static or adaptive streaming, pacing and so on. Hence, the system needs to be agnostic to the details of both the video itself but also how it is delivered. Our solution to this problem is to normalize the metrics we collect (Section 4).

Third, while we found that a multiple vantage point solution is very effective, it may not always be possible to obtain data from all vantage points of interest for reasons such as security or privacy concerns. To address this challenge, we designed our system so it can also diagnose problems, albeit with lower accuracy, if it lacks data from some vantage points. We also iden-

tify for each vantage point how much its measurements contribute to the diagnosis (Section 5).

Finally, there is a growing demand for encrypted content delivery, even for video. This poses a challenge for systems that rely on packet inspection or HTTP traffic analysis. Our design avoids such analysis so it can be compatible with encrypted traffic. We similarly avoid dependencies on any particular wireless technology to ensure wide applicability.

3. SYSTEM DESIGN

The proposed framework consists of one or more *probes* along the data delivery path that provide a number of performance metrics, and a QoE *detection system* that constructs the necessary features from these metrics and applies machine learning algorithms in order to extract the root cause.

3.1 Probes and Metrics

Ideally, each network device along the data path may provide performance indicators. However in practice this is not feasible as it involves the cooperation of too many parties and hardware vendors. Therefore, in our approach we focus on three key vantage points: *i) the mobile device, ii) the connection gateway (e.g., wireless router) and iii) the content server*. These three points can capture issues at the boundaries of each of the three segments in the video delivery path: the user, the ISP and the content provider.

The probes collect performance metrics from all relevant layers:

- **Application layer:** At the mobile probe, we capture statistics concerning the QoE of video playback from the mobile OS irrespectively of the video application (our implementation is done on Android). These metrics include the video startup delay, video stalls, frame skips, the status of the buffer, video bit-rates, etc. These are used to construct an estimated *Mean Opinion Score* ([3]) that represents the QoE ground-truth. Notice that while these metrics can indicate the existence of poor QoE, we are not including them as features in the classifier, i.e., they are only used to provide the labeled QoE ground-truth.

- **OS/Hardware Layer:** The hardware metrics provide information about the available resources and the connectivity state at each of the three VPs. For that purpose, we monitor the percentage of load, CPU utilization, the amount of free system memory and so on. At the end of a video flow, aggregated information about each feature is returned (e.g., average, minimum, maximum, standard deviation of CPU usage).

- **Transport layer:** A set of 113 network metrics are collected per flow, including RTT, number of packets, flow duration, window size, out-of-order and re-transmitted packets, etc. These metrics are collected on all probes for each of network interface using `tstat` [4]. Extensive documentation of these metrics can be found in [5].

- **Link/Physical layer:** For each of the available network interfaces (NICs) the probes extract information about the utilization, bandwidth, and dropped or retransmitted packets. In addition, for wireless links (WiFi/3G), the radio technology, the advertised rate and signal strength information (RSSI) for each of the connected devices is monitored.

  Similarly to the OS/hardware metrics, an aggregated set is calculated for the conditions of each NIC during a video flow. For instance, the average/minimum RSSI or the number of disconnections/handovers during the flow is returned.

Our proposed multi-VP approach enables each entity with a deployed probe to diagnose problems within its own proximity, separately without requiring information from other contributors. This way, providers or users are not limited by common privacy concerns or collaboration restrictions. However, combining information from all three entities can improve root cause analysis accuracy.

3.2 Detection System

Our system uses machine learning to learn the correlations between performance and QoE metrics and to create a model for detecting and characterizing the root cause of playback problems. Before applying the ML tools, we employ two techniques, Feature Construction (FC) and Feature Selection (FS) that help improve the classifier's performance.

Feature Construction aims in making the system more agnostic to the specifics of each scenario, i.e., video type, streaming techniques and network technology in our case. With this method, our objective is to make the prediction model as generalizable as possible so that it can be successfully applied for different devices, video players and video services but also for different network conditions.

To achieve this task, we normalize the features that depend on any of the aforementioned variables. Specifically, we normalize all the parameters which are expressed in bytes or packets with the respective total number of bytes or packets of the entire session. The list of normalized features includes among others, the number of data packets, data bytes, re-transmitted packets, re-transmitted bytes and out of order packets [5]. The same approach is applied for the video duration which is

normalized with the total duration of the video session.

Furthermore, we calculate the uplink and downlink utilization of each device's NIC by dividing the average transfer rate of a video session by the maximum transfer rate observed for this NIC in the entire dataset. In this way, the utilization takes values between zero and one.

The RSSI is collected in one second intervals and then the average, maximum and minimum values are calculated for the entire session. For our analysis we keep the average value only as we observed that it has better predictive capabilities as compared to the maximum and minimum.

Feature Selection: To increase the performance of the algorithm in terms of both accuracy and execution time, it is important to significantly reduce the feature space size. The reduction of the number of features used to train the algorithm, minimizes the over-fitting problem that is either caused by multiple features with little or no predictive power, or by features that contribute the same information to the prediction. After experimenting with different FS algorithms, we find that the Fast Correlation-Based Filter algorithm is the most efficient in identifying a minimal set of features with high predictive power.

After applying FS, the number of features is reduced from 354 to 22 (Table 1). Among the remaining features, those with higher weights were the utilization of the interfaces, the 3 hardware metrics from the mobile device: the free memory, the CPU utilization and the RSSI. In section 5.4 we discuss how much each of these features contributes to identifying individual problems and the improvements resulting from both FS and FC.

mobile CPU utilization	mobile bytes retransmitted
mobile free memory	router out-of-order pkts
mobile RSSI	server avg RTT
mobile downlink utilization	mobile first packet arrival
router downlink utilization	router first packet arrival
server uplink utilization	server max window size
mobile pkts retransmitted	mobile min MSS
server min MSS	mobile max RTT
server video data pkts	router video data pkts
mobile max window size	router reordered pkts
router avg RTT	router max RTT

Table 1: Features after Feature Selection.

Machine Learning: For the data processing and analysis we use version 3.6.10 of Weka. Our classifier of choice for the data analysis is J48 which is an implementation of the popular C4.5 algorithm. The training and testing of the algorithm is performed using the 10-fold cross-validation method.

C4.5 and Decision Trees in general, are known to perform well with noisy data. Therefore, they are a suitable solution for building our predictive model since we intent to train and test it on network data where noise is induced by background variations. We further discuss background variations in Section 4.2.

Decision Trees outperformed other algorithms like Naive Bayes and Support Vector Machines which we also evaluated with our datasets. Given that the datasets from our experiments consist of a large number of features that often have a non-linear relation between them, decision trees are well suited for our predictive model since their performance is not affected by such non-linear relations, while their hierarchical structure fits well with our troubleshooting approach.

Moreover, data collected from real networks can be noisy due to background variations generated by multiple sources. For that reason, Decision Trees are a good solution since they cope well with noisy data.

Finally, another advantage of using C4.5 is that contrary to algorithms such as SVNs, the model is not a black box. The constructed tree can be visualized and interpreted. This can greatly simplify and improve the feature selection process and help optimize the performance of the model.

4. COLLECTING THE GROUND TRUTH

In order to build and train our ML model, we need to collect the ground-truth: a set of labeled good and problematic video instances with a known root-cause. This data set will be used to train our model and also for controlled experiments. Finally, we will use our model in real-world experiments to evaluate how well it can cope with the added noise and complexity of the Internet. To achieve this, we implemented a testbed infrastructure with four components i) a realistic hardware setup with multiple simulated mobile devices and backbone connections, ii) background workloads for generating constant variations, iii) induced impairments that will simulate a specific scenario/label and iv) associate mean opinion score to the collected measurements. To make the generated model as realistic as possible, the settings of each component (e.g., loss rate, link speeds, load, etc) is based on distributions that were derived from traces that were acquired from a network within a large European ISP.

4.1 Setup

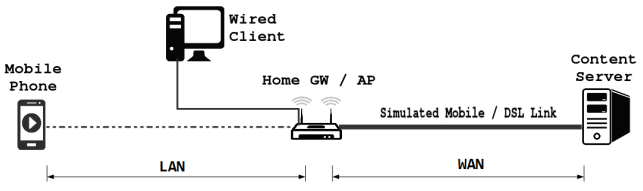

Figure 2: Device setup in the testbed.

We set-up a simple testbed with a video server, a router/AP (Access Point) and three different Android

Simulated Problem	Tools/Method	Settings/Comments
LAN Shaping	`tc`, `netem`	LAN: BW cap=1-70Mb/s, 1ms delay, 0% loss
WAN Sahping	`tc`, `netem`	DSL: BW cap=7.8Mb/s, 50ms delay, 0.75% loss Mobile: BW cap=5.22Mb/s, 100ms delay, 1.4% loss
LAN Congestion	`iperf`	UDP traffic from wired client to the router
WAN Congestion	`iperf`	UDP traffic from wired client to the server
Mobile Load	`stress`	CPU, memory, IO, disk workloads
Poor Signal Reception	distance, attenuation	Reduced SNR and data rate
WiFi Interference	external interference source	Transmission on the same frequency

Table 2: List of simulated problems, used tools and configurations

devices. The phones are connected to the Wireless LAN of the AP and the server is connected via an Ethernet cable to the router. In addition to the devices that are necessary to deliver the video, other wired and wireless devices are available in order to generate background traffic on the network segments and interference on the wireless link (Figure 2).

We use an Apache server to deliver the video. We downloaded the videos from the top 100 most viewed list [6] from YouTube to the server in either Standard or High Definition to ensure the diversity of the video collection. A Netgear WNDR3800 running OpenWRT was used as a router/AP. It was configured to operate at 5GHz after verifying that there are no surrounding sources of interference. Three types of Android devices were used as mobile clients: Samsung Galaxy S II, Nexus S, and Nexus 5. The devices are instrumented with our developed application, which is responsible for performing HTTP video requests to the server and opening the returned video stream using the default Android media player.

	BW	delay	loss
DSL	7.8Mbit/s	50±20ms	0.75±0.5%
Mobile	5.22Mbit/s	100±30ms	1.4±1%

Table 3: Configuration of simulated links

In order to make the generated model as realistic as possible, `tc` and `netem` are used to simulate a DSL and a mobile link with the settings shown in Table 3. The delay and loss for both configurations follow a normal distribution within the indicated ranges. As mentioned in the previous part, these settings were obtained by analyzing traces from a real deployment in a large ISP.

4.2 Background Variations

To recreate realistic network conditions, we introduce synthetic competing traffic workloads of different patterns. These background variations are based on real world network traces and will aid in training the algorithm for successful deployment in the real world. This is done using the D-ITG generator [7], which supports traffic generation based on different applications such as Telnet, FTP, gaming, VoIP and more. We also use ApacheBench to create a realistic load on the server.

4.3 Simulated Problems

In order to generate the dataset that contains various levels of QoE, we iterate through a set of scenarios in which we stream a randomly picked video and artificially induce a problem with varied intensity. Apart from background variations, we use problems in three categories: networking, device hardware and wireless medium issues. The list of simulated problems, the used methodology, and the specific configurations can be found in Table 2.

Shaping and Congestion. To simulate LAN congestion, we use multiple `iperf` instances to transmit UDP traffic between the wired LAN client and the router; for WAN congestion we generate traffic with the same method but between the server and the router.

For traffic shaping, different bandwidth, delay and loss restrictions are applied to the corresponding link. The LAN is shaped based on the data rates offered by common 802.11 standards such as a, b, g and n that are capable of providing rates per stream ranging from 1 up to 70Mbit/s. For the WAN shaping we set different restrictions for mobile and DSL connections (Table 2).

Mobile Load. This category examines cases where the high load on the device hardware does not allow the proper decoding and playback of the video. The load simulation is performed with the workload generator tool `stress` that allows CPU, I/O, memory and disk workload generation.

Poor Wireless Signal Reception. We simulate poor signal reception by placing the phone far from the AP and by attenuating the transmitted signal at the AP. As a result, there is degradation in the wireless link's SNR and the available data rate.

WiFi Interference. This scenario, involves creating interference on the wireless channel from external sources. In real use cases, interference can be caused by near-by devices transmitting or receiving on the same

frequency range. In our experiments interference is created by generating large traffic workloads on an adjacent second WLAN operating on the same channel as the AP we use for measurements.

4.4 MOS-based Labeling

Before performing the analysis, the instances in the dataset need to be labeled with the QoE ground truth so they can be used for training and evaluation of the classifier. QoE labeling has to express the quality of the video session in terms of user satisfaction so that we can correlate problematic videos with the QoE.

For that purpose, we convert application performance metrics such as startup delay and the frequency and duration of stalls to Mean Opinion Score (MOS) ratings based on the work of Mok et al. [8] who derived an equation for calculating the MOS from performance metrics by means of regression analysis. Based on the obtained scores, we label instances with MOS greater than 3 as 'good', instances with scores between 2 and 3 as 'mild' and those with MOS lower than 2 are labeled as 'severe'.

For the detection of the location of the problem, we create six new labels based on the combination of the segment that the issue occurs and its severity. For the evaluation of the algorithm when detecting the exact problem, we label problematic instances according to the type of the fault.

5. EVALUATION

In this section, we evaluate the system's performance in the controlled environment described in Section 4 for detecting the existence of problems, detecting the problem's location and for identifying the exact problem. Later we will examine if the resulting model is robust enough to detect problems in the real deployment.

The training and testing of the algorithm in all the evaluation scenarios is performed using 10-fold cross-validation. We present the system's performance in terms of overall accuracy, defined as the percentage of correctly predicted instances, i.e., the number of True Positives (TP) and True Negatives (TN) over the total number of instances. In addition, we also use the Precision and Recall metrics. *Precision* is expressed by the ratio of TP over TP and False Positives (FP) and represents the accuracy a certain class is predicted. *Recall* is the ratio of TP divided by the total instances in this class and it measures the classifier's ability to correctly identify the desired classes from the data set. In simple terms, for a root-cause c (e.g., low RSSI), high precision means that the framework did not miss-classify other problems as c, while high recall means that it found most of the instances that exhibited c and, therefore, has a high probability of detecting this issue.

The collected dataset consists of 354 metrics including network metrics, the total number of rebuffering events, device CPU and memory utilization and the RSSI. Note that the rebuffering events are only used for labeling the instances and not as a feature. Overall, there are 3919 instances in total out of which 3125 are labeled as good, 450 as mild and 344 as severe.

5.1 Who Can Detect the Existence of a Problem?

First, we examine which of the VPs (or which combination of them) is performing better when identifying the *existence* of a problematic video flow. For that reason, we aggregate all labels into three categories: 'good', 'mild' or 'severe', as discussed in section 4.

As observed in Fig 3, each one of the vantage points can independently discover problematic sessions with similar accuracy: for the mobile it is 88.1%, for the router 86.4% and for the server 85.6%. Finally, when combining the measurements from all the vantage points, the performance slightly improves to 88.8%. We observe that the mobile phone achieves performance that is as good as the combination of all three vantage points as it is in the position to measure both local (e.g., CPU/RSSI) and remote (e.g., server load, network) issues.

Moreover, although the other two VPs achieve more than 85% accuracy in detecting good instances, they have significant problems to discern between mild and severe problems. In more detail, the system's poor performance for mild problem detection is correlated to the high number of false negatives where the problems are identified as severe and the false negatives where they are labeled as healthy.

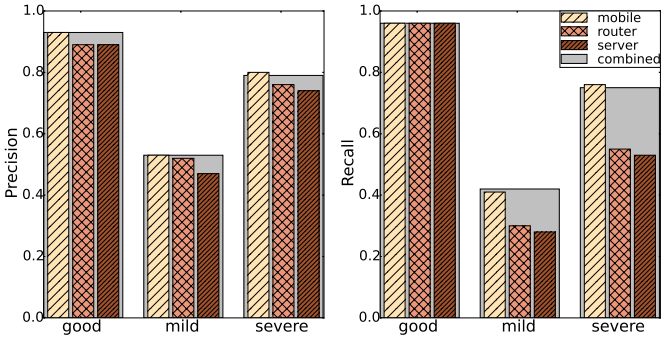

Figure 3: Precision (left) and Recall (right) for problem detection in controlled experiments.

Takeaway: The existence of healthy sessions can be identified with high accuracy from each entity independently. ISPs and content providers can identify that there was a problem but they cannot be certain about its severity in terms of impact on the users' QoE. Moreover, we find that instrumentation closer to the mobile terminals where the majority of the problems occur yields higher performance.

5.2 Who Can Detect a Problem's Location?

Apart from the existence of a problem, it is important for each entity to understand if the fault is within their network or who is to blame when there is an issue. For that reason, we aggregate labels into three categories based on the *location* of the problem: mobile device, LAN and WAN.

What is interesting however, is the ability of the server VP to localize problems in the LAN segment. Specifically, the server shows almost equal performance to the router VP for detecting LAN problems. To better understand why this is the case, we inspect the features that contribute most to detecting LAN problems. We find that for both VPs the same features, RTT, first packet arrival delay and the number of retransmissions, are ranked highest for LAN problem detection.

We also evaluated the benefits of using VP pairs for location detection. However, we did not observe any significant improvement in accuracy nor any intriguing result.

Takeaway: Content providers who deploy such a system have the ability to identify if a problem has occurred on the ISP's network. This information is useful to content providers for spotting congested or underprovisioned ISP networks and pursue better peering agreements with ISPs in order to minimize bottlenecks. ISPs can also identify whether the issue has originated within their own network or the user's LAN if the home router is instrumented.

Finally, an instrumented application or an instrumented phone can provide valuable information to the users to identify whether their home network, their ISP or the content provider is to blame for poor QoE and it can significantly improve the accuracy of the other entities if the measurements are shared.

5.3 Who Can Detect the Exact Problem?

Next, we trained and evaluated the algorithm using all the labels of problematic scenarios that are available in our dataset, allowing us to assess the accuracy with which the classifier can detect the *exact root cause* behind the problem experienced by the user. The overall accuracy for detecting the exact problem, is 88.18% for the mobile VP, 85.74% for the router, 84.2% for the server and 88.95% for all three VPs. These numbers demonstrate the system's high performance when carrying out the task of identifying the root cause behind video QoE issues.

However, while the overall accuracy is good, we observe that certain vantage points exhibit difficulties in discerning certain problems. Figure 4 shows the different accuracy with which each issue is predicted, while Table 4 provides insights about the 3 metrics with the highest prediction power for each label (notice that there are cases where only two or even only one metric make significant contribution).

Furthermore, we observe the high accuracy with which WiFi interference and low RSSI related problems are predicted. From the figure it is clear that all the VPs in the system perform very well when detecting sessions which suffer from severe problems in the wireless medium.

However, the detection of mild interference from the router and the server is done with much lower accuracy. Given that these two VPs don't have RSSI information, they are unable to distinguish the small variations caused by mild interference. As a result, we observe from the classifier's output that a large number of mild interference instances are predicted as healthy witch causes the particular label to be predicted with lower accuracy.

More information that help understand this behavior can be found in Table 4, where for the router and the server, the highest ranked features are RTT and the first packet arrival delay which do not offer much information about the performance of the wireless medium.

Table 4 also offers interesting insights in the features with the highest prediction power for mobile load cases. Specifically, when the mobile VP is used, CPU, memory and RTT are the most important for detecting the problem. However, for the router and server the highest ranked feature is the RTT which has an obviously very little information regarding the load of the device. This can be reflected in the low performance of these two VPs for identifying mobile load problems.

Specifically, for network related issues the important metrics are the interface utilization, the RTT and the number of video packets. In wireless medium problem detection the greatest contribution is made from the RSSI when the mobile VP is used and from RTT for the other two VPs.

The router and server VPs have very poor detection capabilities for mobile load issues since the significant features in this case are the device CPU and memory load. The very low accuracy for these problems by the router and server VPs, is a result of the high number of instances that are detected as healthy which in turn has an impact on the number of false positives.

Apart from the mobile load, there are also other cases in Figure 4 such as mild WAN congestion and shaping where both the server and the router VPs show lower detection capabilities. This poor performance is attributed to the large number of miss-classifications of these faults as either LAN congestion and shaping or healthy and thus increasing the number of FP and FN.

However, for the case of the mobile load and WAN congestion, we find that the combined use of the three VPs significantly improves the detection performance. These findings can motivate ISPs and content providers

	WAN CONGESTED	WAN SHAPED	LAN CONGESTED	LAN SHAPED	MOBILE LOAD	LOW RSSI	WIFI IN/FERENCE
M	1st pkt arrival M RTT M out-of-ord. pkts	1st pkt arrival M out-of-ord. pkts	M RTT M Util. M Bytes re-TX	1st pkt arrival M Util. M RTT	CPU MEM M RTT	RSSI M RTT MEM	RSSI M pkts re-TX M Util.
R	R RTT R 1st pkt arrival R Util.	R out-of-ord. pkts R RTT R 1st pkt arrival	R RTT R 1st pkt arrival R Util.	R RTT R 1st pkt arrival R pkts reordered	R RTT	R 1st pkt arrival R out-of-ord. pkts R RTT	R RTT R 1st pkt arrival
S	S Util. S data pkts S RTT	S Util. S data pkts S win size	S data pkts S Util. S RTT	S data pkts S Util. S win size	S RTT	S data pkts S RTT	S RTT S Util.
C	R Util. S Util. M RTT	S pkts re-TX S Util. R RTT	R Util. R 1st pkt arrival M RTT	M RTT R Util. M 1st pkt arrival	CPU MEM M RTT	RSSI CPU MEM	RSSI M pkts re-TX

Table 4: Feature ranking for exact problem detection (M=mobile, R=router, S=server, C=combined)

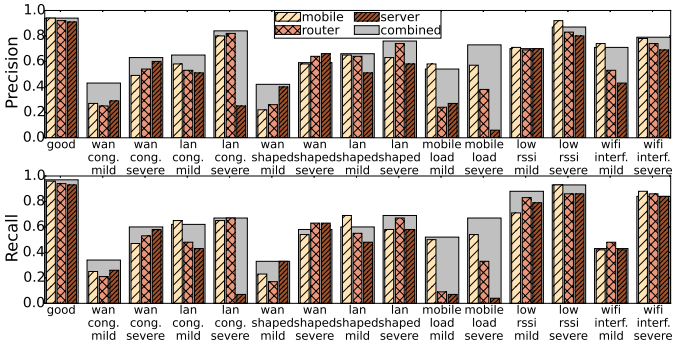

Figure 4: Precision and Recall for exact problem detection per VP.

to pursue collaborations in order to improve the performance.

Takeaway: Each of the entities can independently perform well when detecting a large variety of problems such as the ones related to wireless conditions, severe LAN and WAN congestion and shaping. However, mobile devices are much more accurate in identifying local problems related to high load or wireless interference. Furthermore, there cases such as WAN congestion and mobile load where the combination of all the VPs can significantly improve the performance. Finally, while we observe that all three parties are capable of identifying healthy video sessions with very high accuracy, these results indicate that in order to perform a full-scale accurate root cause analysis, some collaboration between the entities is desirable.

5.4 Which Features Help?

The objective of this section is to illustrate the improvements that can be achieved when using different features to train the model. We evaluate the system with the combination of the three VPs using seven different feature sets, RSSI, hardware metrics, interface utilization, network delay parameters, TCP metrics, all the available features and finally with the set of features after performing Feature Selection (FS) and Feature Construction (FC). For the network delay parameters, we consider all the RTT metrics we have available from the TCP flows.

Figure 5 shows the precision and recall for each of the inputs. When using only the RSSI or only mobile hardware metrics, the accuracy is lower than 35%. Using the interface utilization alone, yields precision and recall values near 55%, while the use of delay alone results in improved accuracies around 70%. The evaluation with the entire feature set further increases the obtained accuracy by 5% but even more improvement is reached when applying FS and FC, with precision and recall values above 80%.

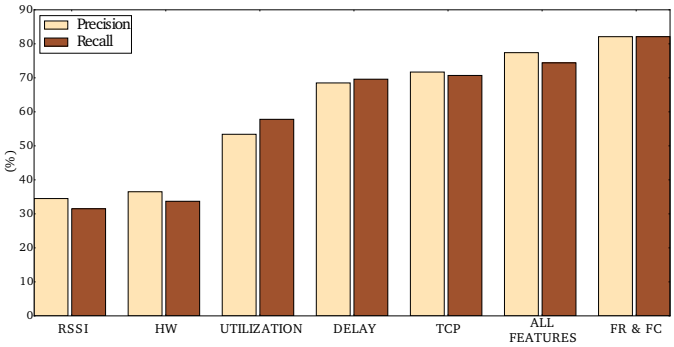

Figure 5: Problem detection accuracy for different feature sets.

From the information provided in Table 4, we observe that the utilization of the interfaces contributes significantly in the detection of the majority of problems. This result highlights the important role that feature construction plays in the problem detection capabilities of the system.

Moreover, one of the features that is used by almost all VPs for predicting congestion and shaping issues, is the first packet arrival time. This metric is an indicator of video sessions with longer startup delays but it is also correlated with network issues such as delay and loss.

The metrics with higher predictive power for mobile

load are the CPU and memory utilization. For the server and the router VPs where these metrics are not available, RTT is used instead but with very poor results as shown in Figure 4, as it does not hold information regarding the device's hardware state.

Takeaway: The results indicate that the RTT and link utilization, as measured by each vantage point are key metrics in performing RCA. Hardware and NIC metrics can further help us to separate individual local issues. Moreover, it is evident that feature construction and reduction plays a significant role in improving the system's accuracy as constructed features are highly ranked by the classifier.

6. REAL WORLD EXPERIMENTS

In this section we describe and discuss the results of the system's evaluation in two real world settings. In the first environment, clients are in a corporate WiFi network where we can artificially introduce faults. In the second case, clients access videos over a wide range of wireless networks including both 3G and WiFi, where faults are not controlled and occur naturally. In both cases, clients retrieve videos from both a private server and YouTube.

6.1 Experiments With Induced Faults

The purpose of the the real world experiments with induced faults, is to get labeled data that will enable us to evaluate the robustness of the trained model on a real wireless network which is characterized by unpredictable topology, constant variations in traffic, signal strength and number of connected devices.

6.1.1 Setup

For the measurements, we distribute five Galaxy S II to equal number of users for a period of one week. The phones are again equipped with an application that automatically launches random videos from the top 100 list, while coordinating the network and hardware probes. The users were instructed to carry the phones with them while inside the wireless range in order to capture variations due to movement and received signal quality.

In these experiments, the videos are streamed from both our private video server and from YouTube with probabilities 0.25 and 0.75 respectively. We select these probabilities so that we end up with a dataset where the majority of measurements corresponds to YouTube sessions and a smaller part to streams from our server. Finally, the phones, the wireless AP and our server were instrumented with probes as described in 3.1.

Using the same methodology as the one in the controlled experiments described in Section 4, we introduce five different types of faults, lan congestion, wan congestion, mobile load, low rssi and wifi interference. Furthermore, we ensure that the conditions of the network allow to successfully load a video just before and after the induced fault. However, since this a semi-controlled environment, we cannot fully guarantee that during each video flow there are not additional (spontaneous) problems over the unmanaged Internet links or video services.

The collected dataset consists of 2619 instances from which 1962 are good, while 463 have mild and 194 have severe QoE issues.

6.1.2 Real-World Evaluation

Our goal is to evaluate the ability of the classifier to predict labels in the real world scenario based on the training that was performed using the controlled dataset.

In this part, we demonstrate the system's capability of detecting *the existence* of problematic instances using either one of the probes or the combination of all three. The detection is done with 88% accuracy when using the mobile probe, 84% when using the router and 81% when measurements from the server probe are only used. The combination of the three probes yields accuracy of 88.1%.

Figure 6 illustrates the Precision and Recall values for this phase of the evaluation. Overall, the results match the controlled experiments. In this case too, the mobile VP outperforms the other two VPs. However, one notable difference is the increase in both Precision and Recall for the mild problem detection. This can be attributed to the fact that the variations and background noise in the current environment is less than the variations we simulated in the controlled experiments.

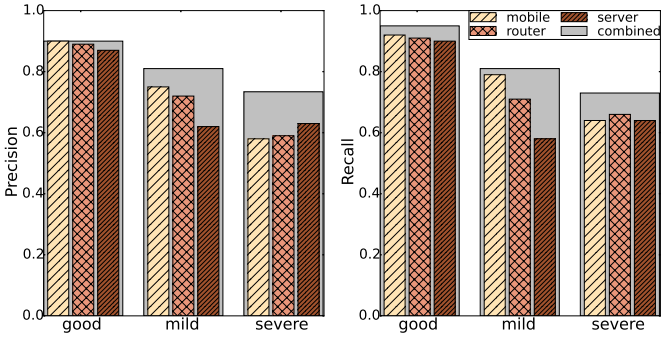

Figure 6: Precision and Recall for problem detection in the real world experiments per vantage point.

Furthermore, we also observe equally good robustness of the trained model in terms of detecting *the exact root cause* of a playback problem. In this case, the combined use of the three vantage points allows correct detection with accuracy of 82.9%. When using separately the mobile, the router and server VP we obtain accuracies

equal to 81.1%, 80.5% and 79.3% respectively.

From Figure 7 we see better performance for device load and wireless medium issues which is to be expected given the strong correlation of these faults with specific hardware metrics. In the LAN congestion scenario we observe better results from the mobile and the router VP while for the case of WAN congestion the server is detecting problems with higher accuracy.

For each of the entities that participate in the video delivery this means that the VP on the client's device is necessary for detecting the root cause of the majority of problems. ISPs on the other hand, can effectively discover LAN faults but also wireless errors such as low RSSI and interference. Finally, content providers can perform WAN fault identification with good accuracy but fall short when it comes to finding faults that occur on the device or in the wireless medium.

Takeaway: Our findings here are in agreement with those in the previous experiments for problem detection and root cause identification. This is a strong indicator that *our system that was initially trained in a fully controlled environment can be successfully applied in the wild*. At the same time, smaller differences in the detection of some problems emphasize the importance of continuous training. While collecting large-scale ground-truth in the wild might not feasible, it is still possible to acquire some labels as specific problems can be recognized by experts within each entity (e.g., network engineers). Furthermore, ground-truth about the quality of experience can be given by means of crowd-sourcing (i.e., people complaining at call centers, or feedback provided by the users within the application).

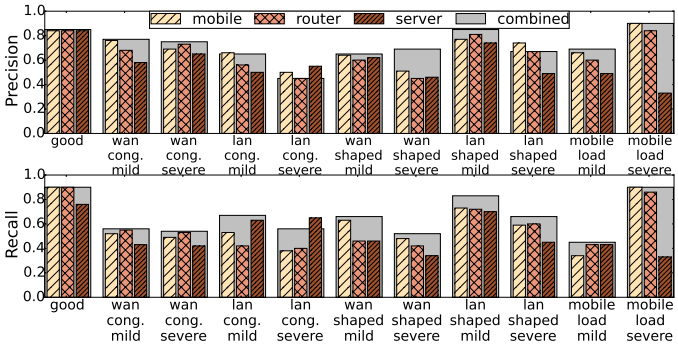

Figure 7: Precision and Recall for problem detection in the real world experiments

6.2 Deployment Without Induced Faults

The final step in the evaluation is *detecting faults that were not induced by us* and, therefore, might be more complex. Furthermore, a particularly important aspect of this evaluation is to test the system in mobile networks, given that there is a constantly growing number of users who watch video over cellular broadband connections.

In this scenario too, we distribute five Samsung Galaxy S II devices to equal number of users for one month with the instruction to carry the phones with them at all times. The phones contain SIM cards with unlimited 3G data-plans, while the users were allowed to connect them to any WiFi access point. This approach allowed us to test the system on a multitude of networks that use either cellular or 802.11 technology.

The videos are again streamed from both our private video server and YouTube with 1:3 ratio so that the final dataset is richer in measurements from the YouTube service. A probe collects network statistics on our video server for the sessions streamed from it. With this methodology we can have three different VP combinations, i) (mobile, router, server) when the user is streaming video from our server while using our WiFi, ii) (mobile, router) when YouTube videos are streamed on our WiFi, iii) (mobile, server) when videos are delivered from our server over other networks and iv) (mobile) when streaming from YouTube on other networks. Given that the majority of the videos were delivered over 3G and in order to make the results comparable between 3G and WiFi, we removed any features from the router (therefore only the mobile and server vantage points are used).

Similar to the previous scenario we use the trained model from the controlled experiments. For the real-world experiments, although all mobile-based measurements (e.g., hardware as well as the number of re-buffering events) are always available, the number of other metrics varies depending on the number of VPs that were used. The real-world dataset contains 3495 instances from which 2940 are good and 555 problematic.

6.2.1 Does it Work in the wild with real faults?

Since the experiments are done in the wild, we cannot obtain the ground truth for the root cause behind the stalls, only the ground truth for stalls and loading time. Therefore, we can only mark instances as good or problematic.

In terms of identifying the *existence* of a problem, the mobile probe, server and their combination still achieve a high accuracy and recall, as shown in Figure 8.

Similar to the controlled experiments, we find that the mobile VP is a better choice than the server for identifying both good and problematic instances, while the combined use improves the system's accuracy.

Takeaway: The results from the real world evaluation verify that the system is equally effective when detecting problems in the wild even when fewer VPs are available. This also reveals that the system can capture successfully cases of mobility although they were not covered in the training phase and it can cope with the

diversity of mobile networks.

A closer look at the results shows that the detection of healthy video sessions is achieved with high accuracy, there is some loss regarding the identification of problematic videos. This loss occurs due to differences in the characteristics of the faults that we encounter in the real world as compared to the ones we induced manually in the previous sections. This effect can be minimized by introducing more VPs (e.g., on 3G RNCs) in order to get more fine grain information about how smaller variations affect the video QoE and by furthermore training the classifier with a wider range of problems. Finally, as discussed in the previous section, these figures are likely to be improved once more labeled faults are fed into the training set.

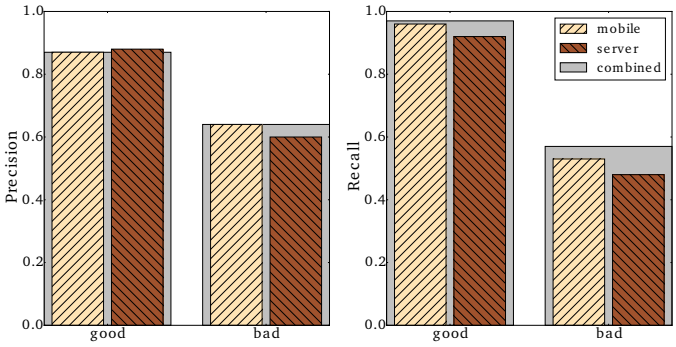

Figure 8: Precision and Recall for problem detection per VP pair in the real world.

6.2.2 *Identifying the Root-cause*

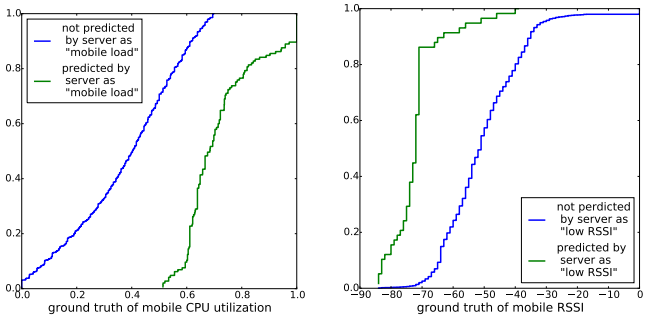

Figure 9: Comparing the server estimations about CPU (left) load and RSSI (right) to the ground truth

We can use the trained model from the controlled experiments to predict the root cause of faults that occurred in the problematic sessions. The results of the predictor's output can be found in Table 5. As we observe, the most common type of problems occur within the users' local network (13% of all instances). Surprisingly only few (2%) of the instances are estimated to be caused by low RSSI or WiFi interference as typically the videos fail to even start a TCP flow when there is very low signal. Furthermore, a number of instances (4%) were problematic due to an estimated high mobile load.

As discussed in the previous section, we can directly calculate that the algorithm correctly identifies good instances with 85% accuracy. Furthermore, although it is not possible to verify all of these estimated root causes, we still have the ground truth for some of them: mobile load and low RSSI.

Figure 9(left) shows the distribution of CPU load on the mobile device for problematic videos sessions as predicted by the video server Vantage Point. Two different distributions of the CPU ground truth are given: video sessions that server VP labeled as high "mobile load" and the remaining video sessions. The results show that, although the server vantage point only has access to transport layer metrics (TCP statistics), the video flows that were estimated as high mobile load have indeed much higher CPU utilization.

Similarly, Figure 9(right) shows the distribution of RSSI for the instances that were considered as low RSSI from the point of view of the server's vantage point. As before, we observe that the server vantage point can successfully identify these instances despite the fact that the phones were connected to various WiFI and 3G networks.

Takeaway: These results further reinforce our hypothesis that a model that was trained in a controlled environment is robust enough to be applied as a starting point on a real world environments where the network conditions and the faults can be highly dynamic and unpredictable. Furthermore, we observed that even the service provider VPs can identify problems that occurred within the users network or device (e.g., low RSSI or high CPU usage) without any external information.

GOOD	WAN CONG.		LAN CONG.		MOBILE LOAD		LOW RSSI		WIFI INTER.	
	M	S	M	S	M	S	M	S	M	S
2499	163	166	18	446	2	132	26	0	43	0

Table 5: Real-world root cause predictions (M=mild, S=severe)

7. PRACTICAL IMPLICATIONS

For the end user, our results indicate that even an isolated mobile application that collects measurements from multiple layers can successfully identify a large number of problems without further instrumentation. Such a system can be a powerful tool towards diagnosing video playback QoE issues and where they have occurred. Therefore, when the user is made aware of the location of the problem and if it is originating from the

local network or the device itself then troubleshoot it. Otherwise, the issue can be reported to the responsible entity to take the necessary action.

For the ISPs, the results demonstrate that they can also independently identify problematic sessions, even when traffic is encrypted. Furthermore, they can identify if the problems originate within their own network, in order to fix problematic segments and bottlenecks, but also guide users to solve problems in their home and/or their devices.

For content providers, there are deployment benefits such as detecting loaded servers and network segments in their CDN if the problem occurs on their side, or adapting the content for problematic connections without instrumenting the client when the problem is originating from either the user's or the ISP's side. This is a valuable tool when identifying SLAs and net neutrality violations.

Collaboration: As we showed in the majority of the controlled and real world evaluation scenarios, there are significant improvements (in terms of identifying all possible problems) when two or more entities collaborate to troubleshoot QoE issues. The greater benefits however, are obtained for the entities which collaborate with the end users, since the mobile device has access to valuable information of the local hardware and network performance. This calls for instrumented players or mobile devices.

At the same time, as collaborations might not be possible, an iterative root cause analysis might be employed where each of the entities independently perform analysis within their own infrastructure. Then they report to the other entities along the path whether or not the problem has occurred in their segment. In this way, no sensitive information is exchanged among users or providers, collaborations can be easier established and the deployment of the system can span over a wider range of networks and devices.

Continuous Training: Once the system is deployed in one or more entities, its fault detection and root cause analysis capabilities can be further improved by means of continuous training. This can be achieved by manually labeling new instances based on the observed problem and feeding this information back to the training model. As new data is being added to the training set, the system's accuracy will continue to improve.

One of the limitations of our system, is the inability to detect faults that it has not been trained for yet in the lab. These would not only include new problems such as middleboxes and DNS or routing miss-configurations but also the co-occurrence of problems that jointly affect video QoE.

8. RELATED WORK

Path Diagnosis: The works presented here deal with common issues in wireless, broadband and WANs. This information provided useful insights for the problems that may affect the performance of video streaming services and the users QoE.

In [9] intra- and inter-ISP links were measured to identify issues affecting video streaming QoE. The findings show that most of the issues originate from fluctuations in intradomain links, however there is no clear correlation of these problems with QoE. Finally, [10], showed that voice streaming over backbone links is only affected in rare cases of packet loss.

Mobile Video Traffic Characterisation: In [11], the authors analyse YouTube traffic from a university campus network to conclude that caching improves the performance and the scalability of the service. Plissonneau et al. [12], study the impact of throughput and delay on YouTube abandonment for DSL users. In [13], distributed active measurements are used to measure YouTube and find the effect of redirections and load balancing on video performance. The authors of [14] propose a YouTube traffic generation model based on traces collected from real use cases.

Plissonneau et al. [15] analysed the performance of video streaming over 2G and 3G, while a more recent work [16] evaluated the impact of YouTube on mobile networks. [17] reported 10% packet loss due to redundant TCP connections when streaming on Android and iOS mobile devices. Hoque et al. [18] studied the energy consumption with five mobile video streaming services. [19] provided a comparative study between Android and iOS video streaming where larger number of duplicate data was found on iOS.

The information in the works mentioned above, allow us to obtain a more concrete understanding of the generated traffic patterns and important parameters that affect the performance of these services.

Video Streaming QoE and QoS Correlation: Krishnan et al. [20] used quasi-experimental designs correlate the abandonment rate with the startup delay or the total buffering time. In [8] the authors concluded that the main metric affecting the QoE is the rebuffering frequency. Dobrian et al. [3], show that abandonment is affected by the buffering ratio and startup time.

In [21] a predictive model for video QoE is used to improve user engagement by 20%. The same author in [22] employed machine learning to predict user engagement. In [23], user behaviour is correlated with startup delay, redirections and server response time. Schatz et al. [24] used passive network measurements at ISP-based VPs to infer the rebuffering frequency and duration.

Contrary to these works, our system does not aim at improving user engagement nor at estimating the video QoE from QoS metrics. We focus on identifying video sessions with low QoE scores and accurately detecting the location and the root cause of the problem.

9. CONCLUSIONS

In this paper we presented a multi-vantage point system for detecting video QoE issues and identifying their root cause. Our approach utilizes performance metrics from multiple layers which makes it agnostic to video characteristics and streaming mechanisms but also to encrypted traffic. With the aid of feature reduction and construction techniques, the detection and RCA of problems is done with a minimal set of performance metrics while we ensure that the methodology is generalizable and can be applied to different video services and clients. We further showed that each of the entities which contribute to the video delivery is capable of detecting poor QoE and identify underlying faults without having to share information with other parties.

The next step in this work, is to extend the list of problems that can be identified and train the system for multi-problem detection. To further improve the system's accuracy, we will examine dividing problematic sessions into more labels in order to obtain a more fine grain classification of the severity of the problem.

10. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 318627 (Integrated Project "mPlane") and partially funded by the Spanish Ministry of Economy and Competitiveness under contracts TEC2011-27474 and TEC2014-59583-C2-2-R, and by AGAUR (ref. 2014-SGR-1427).

11. REFERENCES

[1] "Cisco Visual Networking Index: Global Mobile data Traffic Forecast Update 2014-2018".

[2] "YouTube Statistics". https://www.youtube.com/yt/press/statistics.html.

[3] Dobrian F. et al. "Understanding the Impact of Video Quality on User Engagement". *Proc. of SIGCOMM*, 41(4):362–373, 2011.

[4] "Tstat, TCP STatistic and Analysis Tool". http://tstat.polito.it/index.shtml.

[5] "Tstat Log Files Documentation". http://tstat.tlc.polito.it/measure.shtml#log_tcp_complete.

[6] "Most Viewed Non-Vevo / Non-Music Videos". https://goo.gl/MHKpV4.

[7] Botta A. et al. "A tool for the generation of realistic network workload for emerging networking scenarios". *Computer Networks*, 56(15):3531–3547, 2012.

[8] Mok R. et al. "Measuring the Quality of Experience of HTTP Video Streaming". In *Proc. of IM*, pages 485–492, 2011.

[9] Venkataraman M. et al. "Quantifying Video-QoE Degradations of Internet Links". *TON*, 20(2):396–407, 2012.

[10] Markopoulou A. et al. "Assessment of VoIP Quality over Internet Backbones". In *Proc. of INFOCOM*, volume 1, pages 150–159, 2002.

[11] Gill P. et al. "Youtube Traffic Characterization: a View from the Edge". In *Proc. of SIGCOMM*, pages 15–28, 2007.

[12] L. et al. Plissonneau. "Revisiting Web Traffic from a DSL Provider Perspective: the Case of YouTube". In *Proc. of the ITC specialist seminar*, 2008.

[13] Adhikari V. et al. "Vivisecting YouTube: An Active Measurement Study". In *Proc. of INFOCOM*, pages 2521–2525, 2012.

[14] Ameigeiras P. et al. "Analysis and Modelling of YouTube Traffic". *Transactions on Emerging Telecommunications Technologies*, 23(4):360–377, 2012.

[15] L. et al. Plissonneau. "Mobile Data Traffic Analysis: How do you Prefer Watching Videos?". In *Proc. of ITC*, pages 1–8, 2010.

[16] Ramos-Muñoz J. et al. "Characteristics of Mobile Youtube Traffic". *Wireless Communications*, 21(1):18–25, 2014.

[17] Nam H. et al. "A Mobile Video Traffic Analysis: Badly Designed Video Clients can Waste Network Bandwidth". In *Globecom Workshops*, pages 506–511, 2013.

[18] Hoque M. et al. "Investigating Streaming Techniques and Energy Efficiency of Mobile Video Services". *arXiv:1209.2855*, 2012.

[19] Liu Y. et al. "A Comparative Study of Android and iOS for Accessing Internet Streaming Services". In *Passive and Active Measurement*, pages 104–114. Springer, 2013.

[20] Krishnan S. et al. "Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs". *TON*, 21(6):2001–2014, 2013.

[21] Balachandran A. et al. "Developing a Predictive Model of Quality of Experience for Internet Video". In *Proc. of SIGCOMM*, pages 339–350, 2013.

[22] Balachandran A. et al. "A quest for an Internet Video Quality-of-Experience Metric". In *Proc. of HotNets*, pages 97–102, 2012.

[23] Finamore A. et al. "Youtube Everywhere: Impact of Device and Infrastructure Synergies on User Experience". In *Proc. of SIGCOMM*, pages 345–360, 2011.

[24] Schatz R. et al. "Passive YouTube QoE monitoring for ISPs". In *Proc. of IMIS*, pages 358–364, 2012.