2018 30th International Teletraffic Congress

Deriving YouTube Playout Phases from Encrypted Packet Level Traffic

Stefan Geissler*, Stanislav Lange*, Florian Wamser*, Tobias Hoßfeld*
*University of Wuerzburg, Germany
Email: {stefan.geissler|stanislav.lange|florian.wamser|hossfeld}@informatik.uni-wuerzburg.de

***Abstract*—From the point of view of telecommunication providers, video streaming is one of the most demanding applications in today's Internet. Over 73% of the total global network traffic has been attributed to video streaming applications in 2016. In this work, we provide a first step towards a better understanding of the packet level behavior of video streaming traffic to enable more efficient traffic engineering in the future. We perform a measurement study with the popular video streaming platform YouTube and show that the different playout phases of a video streaming session can not only be observed by evaluating application layer metrics, but also from raw and encrypted packet level traces.**

I. Introduction

Video streaming is now the most popular service on the Internet. Most of the world's Internet traffic is generated by video streaming [1]. Worldwide, video traffic will account for 82% of all Internet traffic by 2021, compared to 73% in 2016 [2]. With more than 1 billion users per month, YouTube is the largest contributor to network traffic in the world, used by almost one-third of all Internet users [3]. YouTube alone contributed over 20% to the global network traffic in 2016 [3]. This ever growing demand for video streaming puts pressure on telecommunication providers that need to offer suitable services to both businesses and customers [4] while combating ever falling revenues [5], especially when it comes to mobile video streaming. This shows the importance of efficient traffic engineering when it comes to traffic streams generated by video streaming applications.

Many studies have been conducted in order to characterize the behavior of video streaming applications [6]–[11]. However, most studies are based on application layer metrics to monitor the health and behavior of a player session [7], [8], [10], [12], [13]. As this data is in general not available to telecommunication providers, the goal of this paper is to evaluate characteristics obtained directly from the encrypted packet stream in the network. To this end, we capture incoming and outgoing packets at a video streaming client and extract several metrics such as the inter-arrival time of requests in order to identify the health and state of a video streaming session without the need for application layer metrics. Overall, this work provides a detailed and in-depth study of raw and encrypted packet level traces generated by the popular video streaming platform YouTube. We show that well known characteristics of video streaming sessions can also be detected at a packet level. Our evaluations demonstrate that the identified playout phases differ significantly from each other, while exhibiting similar characteristics between different videos and experiment repetitions.

The remainder of this paper is structured as follows. Section II provides background information regarding DASH video streaming mechanisms as well as related work from the field of video streaming. Following, Section III describes the testbed used to perform the measurements presented in this work before Section IV discusses obtained results. Finally, Section V summarizes our findings.

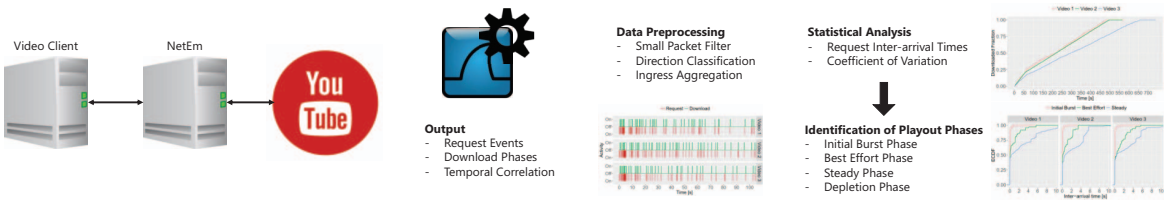
II. Background and Related Work

This work presents a study of YouTube streaming behavior at the network level. The prevalent streaming technique for video-on-demand (VoD) streaming portals is HTTP Adaptive Streaming (HAS) as standardized in Dynamic Adaptive Streaming over HTTP (DASH) [14] and HLS [15]. YouTube has been using HTML5 as the default playback option since early 2015, and not Flash, which was previously used. As part of this, YouTube also uses DASH in HTML5 wherever possible (e.g., IE11, Chrome, Safari) [16]. Netflix also uses MPEG-DASH [16].

The streaming technology defines the *traffic pattern* at network level through its video requests and determines the network traffic [7]. Basically, the video player asks on client side through included adaptation algorithms the video data over the network according to its buffer level or throughout estimation. To make streaming cost effective, it only requests the data it needs to play out, with the exception of a certain amount of pre-buffered data in the application buffer. This defines the *network request pattern* for DASH streaming in correlation with the common streaming phases in video player that are namely (1) initial downloading for buffer filling (filling phase), (2) steady phase in which a relatively constant buffer level is maintained by the video player, and (3) the depletion phase when the buffer runs empty at the end [7], [17].

There are many works that look at application layer metrics like streaming phases and adaptation algorithms. However, few characterize network layer metrics in relation to DASH streaming. The investigation of network layer metrics can provide additional insights into the type of load DASH video streaming imposes on a network. In the following we subsume relevant related work in detail. In [7], authors introduce a model of YouTube considering transport, application, and user level for fixed video qualities. In particular, they introduce a

978-0-9883045-5-0/18/$31.00 ©2018 ITC
DOI 10.1109/ITC30.2018.00023



(a) Testbed Setup. (b) Data Preprocessing. (c) Statistical Analysis.

Fig. 1: Summary of the evaluation workflow from measurement to phase identification.

network traffic model for the YouTube flow control mechanism, which permits to understand how YouTube provisions the video traffic flows to the users. They also investigate how traffic is consumed at the client side, and derive a model for the YouTube application. Finally, they analyze the operation of YouTube from an end-user perspective, presenting a model for the quality perceived by users. Similar models for different aspects can be found in [6]–[11]. In general, the investigations and measurements in literature mainly cover content statistics and video properties such as popularity or video category [6], [18], video content bit-rates [6], [11], user session behavior [6], [18], [19], the video player playback model [7], [8], or the resulting quality of experience (QoE) for the end user [12], [13], [20], [21]. With this work, we want to make a contribution to understanding the network traffic on flow and packet level. Similar approaches have already been evaluated in [22] by using complex machine learning approaches. In this work, we pursue a statistical approach to traffic classification in order to eliminate the need for large training data sets.

III. Testbed Setup and Experiment Design

The data evaluated in this paper has been obtained using the testbed depicted in Figure 1a, which is realized as a virtualized environment running in OpenStack. The testbed consists of three dedicated virtual machines, namely a control instance, a video client, as well as a virtual access gateway responsible for rate limiting.

The testbed controller is not directly involved in the measurements, but is responsible for configuration tasks as well as gathering measurement data. The video client is automated using Selenium[1] and runs a headless Firefox browser. The measurements are conducted by opening a custom HTML document showing only an embedded YouTube video player with a canvas size of 1920x1080 pixels and autoplay enabled. The network traces are gathered using tcpdump running directly on the video client, thereby monitoring all ingress as well as egress traffic. Finally, a dedicated virtual machine works as an access gateway and is used to configure network parameters such as available bandwidth.

Measurements have been performed for three different Videos, namely Video 1: 2d1VrCvdzbY, Video 2: N2sCbtodGMI and Video 3: OHOpb2fS-cM, with a duration of 9:22, 9:20, and 12:15 minutes respectively. For all measurements conducted in this work, a constant bandwidth of 50 Mbit/s has been used. All videos have played out in 1080p and no quality switches have been observed during the presented measurement runs. Each video has been played out and monitored 10 times to evaluate the variance introduced by factors outside of our control during the measurements.

[1] https://github.com/SeleniumHQ/selenium

IV. Results

This section provides the results of the experiments that were outlined in Section III. First, we investigate the processes that correspond to requesting and downloading the video stream. Then, we decompose the processes by identifying and characterizing different phases within each session. Finally, we demonstrate that these characteristics are independent of the video source while differing significantly from each other.

Before continuing with an in-depth evaluation of the statistical characteristics of the obtained data, we pre- process the dataset as depicted in Figure 1b. In order to filter out TCP signaling traffic like SYN and ACK packets that are not relevant for our investigations, we remove packets that are smaller than 100 Byte from our data set for the remainder of this work. Additionally, we divide the traffic into incoming and outgoing by annotating whether the IP of our client VM appears in the destination or source IP header field of a packet, respectively. Figure 2 displays incoming packets as red vertical line segments whose position on the x-axis corresponds to their time of arrival. In the case of download traffic, the time series is divided into intervals of 0.1 second duration which are considered “active” or “on” if at least one packet is received during the interval. Each subplot corresponds to one video and the x-axis is limited to a range between 0 and 100 seconds for the sake of readability.

We can observe a temporal correlation between requests and download phases that follow them immediately. Furthermore, the density of requests, and therefore also downloads, is significantly higher in the beginning and decreases after roughly 5 seconds. This behavior is in line with literature on DASH which describes several phases during the playback of one video, in particular an initial burst for filling the buffer [7], [17].

According to Figure 1c, we evaluate statistical characteristics of the different phases identified based on the previous

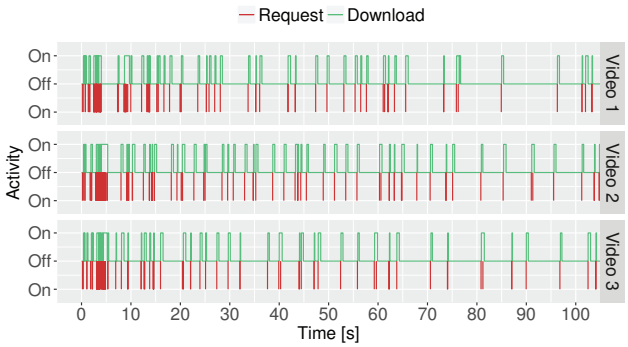


Fig. 2: Request events and corresponding download phases for different videos.

observations. In addition to the qualitative analysis of the video streaming process in the previous figure, Figure 3 shows the development of the amount of downloaded data over time. For a given time on the x-axis, the y-axis denotes the ratio between the cumulative number of bytes that have been downloaded until then and the total number of downloaded bytes during the corresponding experiment run. Differently colored curves denote the three different videos.

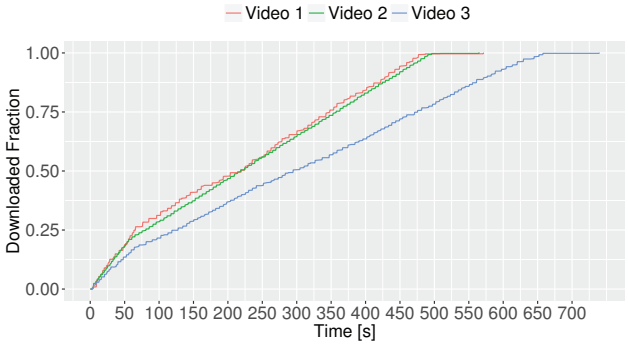


Fig. 3: Fraction of downloaded video data over time for different videos.

We can identify three distinct phases. Firstly, from 0 until roughly 60 seconds, the highest slope is observed. This corresponds to raising the buffer fill level above a desired threshold in a best effort manner. Secondly, a lower, steady slope is observed that ends about one minute prior to the end of the video. During this stage, YouTube's DASH adaptation heuristic attempts to maintain a consistent buffer level that stays within a range that is constrained by a lower and an upper threshold [7], [23]. Finally, in the last stage, referred to as buffer depletion phase, the slope equals 0 since the download is already completed and the remaining video is played out from the buffer.

From the previous two figures, we can derive a total of three phases that we consider relevant regarding the packet level behavior: the initial download burst between the start of an experiment and 5 seconds, the best-effort buffer filling phase until 60 seconds, and the steady phase in which a relatively constant buffer level is maintained. In order to confirm that these phases differ from each other in a statistically significant manner, we present the cumulative distribution functions of the inter-arrival times of requests in each phase in Figure 4. In each subfigure that corresponds to one particular video, differently colored curves represent the three phases.

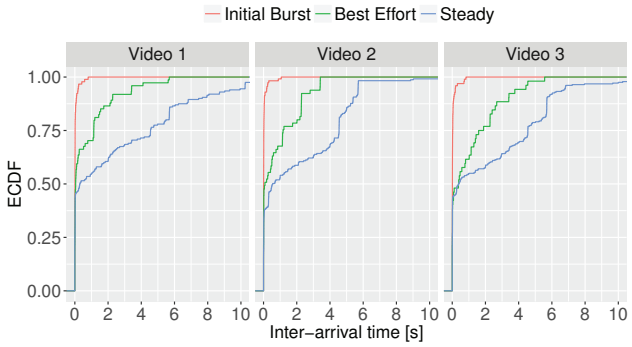


Fig. 4: Empirical CDF for the three different phases and different videos.

For all three videos, significant gaps between the curves of consecutive phases are observed. Additionally, almost no intersections occur between the curves. This indicates that quantiles - and therefore also the corresponding mean values - increase monotonously from phase to phase. Furthermore, Table I displays the mean inter-arrival times of requests alongside their 95% confidence intervals for the three videos and the three different phases.

TABLE I: Mean inter-arrival times of requests in seconds and 95% confidence intervals for different videos and different phases.

	Initial Burst	Best Effort	Steady
Video 1	0.05 ± 0.03	0.88 ± 0.14	2.57 ± 0.03
Video 2	0.04 ± 0.00	0.85 ± 0.02	2.15 ± 0.01
Video 3	0.05 ± 0.01	1.05 ± 0.13	2.38 ± 0.03

Independently of the video, identical phases show a high degree of similarity w.r.t. the mean inter-arrival time of requests whereas the means of different phases vary significantly. While the initial burst phase has the lowest mean inter-arrival times of 40 to 50 ms, the mean time between requests in the best effort phase is in the order of magnitude of one second, and over two seconds during the steady phase. For all videos, the confidence intervals that correspond to different phases do not overlap and therefore indicate a statistically significant difference.

In summary, we demonstrate that different phases during a video streaming session whose existence at application level is acknowledged in the DASH literature can also be observed at packet level. On the one hand, this suggests that it might be possible to identify these phases even despite encryption that prohibits parsing application level information via techniques

like Deep Packet Inspection (DPI). On the other hand, such an approach would have a significantly lower monitoring overhead since only packet sizes need to be measured and payloads can be ignored.

V. Conclusion

In this paper we have made a first step towards a generic traffic classification of network traffic generated by the YouTube video streaming platform. YouTube alone contributes over 20% to the total global amount of network traffic, indicating the importance of understanding the characteristics of its traffic streams. With the aid of network level measurements, we have evaluated if known characteristics of YouTube video streams that can be observed using application level metrics, can also be detected via encrypted packet traces. We have shown that it is possible to distinguish different playout phases from one another through statistical analysis of the inter-arrival times of requests and their corresponding downloads.

Next steps in this direction will include the live detection of these different phases without the information a full reference scenario provides as well as further classification of the traffic characteristics observed in encrypted packet level traces of DASH video streams.

References

[1] Sandvine, "Global Internet phenomena: Latin America & North America," Tech. Rep., Jun. 2016.

[2] Cisco, VNI, "Cisco Visual Networking Index: Forecast and Methodology 2016–2021.(2017)," Tech. Rep., 2017.

[3] YouTube. YouTube Press Room - Statistics. [Online]. Available: https://www.youtube.com/yt/press/statistics.html

[4] Huawei, IHS Markit, "Video as a Core Service for Telcos – Analysis of 50 Leading Operators in Achieving Video Business Success," Tech. Rep., 2018.

[5] The Economist Intelligence Unit Limited, "Telecoms in 2018 - A special report from The Economist Intelligence Unit," Tech. Rep., 2017.

[6] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM, 2007, pp. 15–28.

[7] F. Wamser, P. Casas, M. Seufert, C. Moldovan, P. Tran-Gia, and T. Hossfeld, "Modeling the youtube stack: From packets to quality of experience," *Computer Networks*, vol. 109, pp. 211–224, 2016.

[8] A. Mondal, S. Sengupta, B. R. Reddy, M. Koundinya, C. Govindarajan, P. De, N. Ganguly, and S. Chakraborty, "Candid with youtube: Adaptive streaming behavior and implications on data consumption," in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, 2017, pp. 19–24.

[9] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. M. Lopez-Soler, "Analysis and modelling of youtube traffic," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 4, pp. 360–377, 2012.

[10] A. Rao, A. Legout, Y.-s. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*. ACM, 2011, p. 25.

[11] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 345–360.

[12] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao, "Deriving and validating user experience model for dash video streaming," *IEEE Transactions on Broadcasting*, vol. 61, no. 4, pp. 651–665, 2015.

[13] T. Hoßfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz, "Quantification of youtube qoe via crowdsourcing," in *Multimedia (ISM), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 494–499.

[14] "Dynamic adaptive streaming over HTTP (DASH)," ISO/IEC, Standard, May 2014.

[15] R. Pantos and W. May, "HTTP live streaming," IETF, Tech. Rep., Sep. 2011.

[16] S. Lederer, "Why YouTube & Netflix use MPEG-DASH in HTML5," Tech. Rep., Feb. 2015.

[17] D. Tsilimantos, T. Karagkioules, A. Nogales-Gómez, and S. Valentin, "Traffic profiling for mobile video streaming," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[18] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network–measurements, models, and implications," *Computer networks*, vol. 53, no. 4, pp. 501–514, 2009.

[19] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Characterizing user sessions on youtube," in *Multimedia Computing and Networking 2008*, vol. 6818. International Society for Optics and Photonics, 2008, p. 681806.

[20] C. Sieber, A. Blenk, M. Hinteregger, and W. Kellerer, "The cost of aggressive http adaptive streaming: Quantifying youtube's redundant traffic," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 1261–1267.

[21] C. Sieber, P. Heegaard, T. Hoßfeld, and W. Kellerer, "Sacrificing efficiency for quality of experience: Youtube's redundant traffic behavior," in *IFIP Networking Conference (IFIP Networking) and Workshops, 2016*. IEEE, 2016, pp. 503–511.

[22] D. Tsilimantos, T. Karagkioules, and S. Valentin, "Classifying flows and buffer state for youtube's http adaptive streaming service in mobile networks," *arXiv preprint arXiv:1803.00303*, 2018.

[23] V. Burger, T. Zinner, L. Dinh-Xuan, F. Wamser, and P. Tran-Gia, "A generic approach to video buffer modeling using discrete-time analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2018.