Prediction of Quality Degradation for Mobile Video Streaming Apps: A Case Study using YouTube

Dhruv Jain*, Swapnil Agrawal†, Satadal Sengupta‡, Pradipta De§, Bivas Mitra¶, Sandip Chakraborty‖
*†‡¶‖Department of Computer Science and Engineering, IIT Kharagpur, India 721302
‡Department of Computer Science, State University of New York (SUNY),Korea 406-840
Email: *dhruvjaincse@iitkgp.ac.in,†swapnila@iitkgp.ernet.in,‡satadal.sengupta.nit@gmail.com,
§pradipta.de@sunykorea.ac.kr, ¶bivas@cse.iitkgp.ernet.in, ‖sandipc@cse.iitkgp.ernet.in

***Abstract*—The growing popularity for developing streaming media applications over HTTP triggers new challenges for managing video quality over mobile devices. Quality of online videos gets significantly affected due to the capacity fluctuations of underlying communication channel, which is very much common for cellular mobile networks. Nevertheless, streaming over HTTP makes the video data packets look alike normal browsing data and therefore, the Internet service providers (ISP) have no way to trigger a service differentiation for controlling the video quality. In this poster we explore the traffic characteristics of mobile streaming video apps, by considering YouTube Android app as a use case. We show that by observing the traffic pattern, we can predict possible video quality degradation and video re-buffering events. We develop a methodology for early prediction of possible re-buffering. The experimental results revel that our proposed scheme works with very high accuracy.**

I. Introduction

The increasing penetration of smart-phones, tablets and other mobile devices gradually changing the way end-users consume Internet contents – particularly, the multimedia contents, like audio, video, news etc. The streaming video services have witnessed a shift from conventional delivery via the real time protocol (RTP) or real time streaming protocol (RTSP) to streaming over data service protocols like HTTP or HTTPs [1]. This is mainly due to the popularity of mobile applications, popularly known as "Apps", that use HTTP/HTTPS tunneling for content delivery [2].

Although streaming over HTTP/HTTPs has advantages in terms of data transfer flexibility and network scalability [1], it has several disadvantages as well. First, service differentiation is no longer possible at the Internet service provider (ISP) level, as the video traffic gets a best effort envelope, and therefore looks no different than normal browsing traffic. Second, prediction of end-user's quality of experience (QoE) becomes difficult as the video streaming server does not get any clue about a sudden network capacity degradation, which is very common in wireless and mobile environments.

In this poster, we consider a use case of video streaming through Android YouTube app, and study its data traffic characteristics. We observe that YouTube exhibits different traffic characteristics during normal streaming and when the streaming is affected due to network quality degradation. Based on this observation, we use traffic characteristics as a signature to predict possible degradation in network capacity that may affect normal video rendering, which in turn triggers video re-buffering. It can be noted that re-buffering for streaming media services is an important QoE metric [3] – frequent re-buffering significantly degrades end-user experience. The experimental results reveal that our algorithm can predict possible video quality degradations with high accuracy.

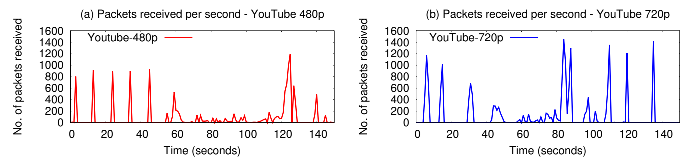

Fig. 1. YouTube Traffic Characteristics at Different Conditions

II. Data Collection

We have collected YouTube traffic traces from 4 different volunteers, through YouTube Android app over Moto-X second generation smart-phone. The data have been collected from both cellular and Wi-Fi networks at different traffic condition with various indoor mobility pattern. During data collection, the videos have been rendered at different fixed resolutions – $240p$, $360p$, $480p$, $720p$ ('p' stands for progressive scans), as well as in the auto-adjustable resolution mode. During the data collection, we have also manually tagged the video quality information, like when there is a re-buffering and auto-degradation of video resolution. This information has been used as the ground-truth for comparing the performance of our proposed mechanism.

Throughout this poster, we use the term "good quality" to indicate that the video has been rendered in a continuous fixed resolution. If the resolution drops during rendering, we call it a "bad quality".

III. Analysis of YouTube Streaming Video Data

YouTube uses an adaptive buffering technique during video streaming, and therefore traffic pattern shows a periodic bursty nature when the network condition is favorable, and the video is of good quality. Fig. 1 shows the traffic distribution of YouTube with respect to time, for two different resolution video rendering – $480p$ and $720p$. We map the traffic pattern with the ground truth of video quality that we logged during data collection, and arrive at following observations:

TABLE I

VIDEO RE-BUFFERING AND NETWORK QUALITY DETECTION ACCURACY (A: ACTUAL, PG: PREDICTED GOOD, PB: PREDICTED BAD)

Length	**Resolution**	**GCSZ (A)**	**GCSZ (PG)**	**GCSZ (PB)**	**BCSZ (A)**	**BCSZ (PB)**	**BCSZ (PG)**	**CZ**
480 sec	720p	28	22	1	2	2	0	5
460 sec	480p	26	24	0	2	2	0	2

1) When the network quality is favorable and the video quality is good, the traffic pattern exhibits periodic bursts of almost similar periodicity and equal burst sizes. From the figure, we observe that the video quality is good in between 0 to 40 seconds for Fig. 1(a) and from 100 seconds to 140 seconds for Fig. 1(b).
2) When the network connectivity is good, the bursts are short but sharp; this indicates that when the burst duration is small but peak size is large, we observe a good quality video.
3) When the channel quality drops that affects the video rendering quality, we see a gradual drop in the peak size of the bursts and the burst duration tends to become wide.
4) During the re-buffering, we observe a wider burst duration, although with a sharp peak size for the bursts. In Fig. 1(a), the burst at time 120 seconds indicates a re-buffering. Similarly in Fig. 1(b), a re-buffering is pointed at the burst at time $80-90$ seconds.
5) Further, when the video quality is good in a favorable network, the peak size and the burst duration has a direct relationship with the video resolution. For instance, with $480p$ resolution, the peak size is 800 packets whereas with $720p$, peak size increases to 1400 packets. Similarly, the burst duration with $720p$ is more compared to the burst duration with $480p$.

Tasks: In case of streaming video apps, a re-buffering occurs because the system fails to predict the sudden drop in network capacity. As a consequence, the video is re-buffered and rendered at a very low resolution. If we can predict the drop in network capacity that may trigger a video re-buffering, the network can send an early notification to the video streaming server to take preventive measures for finding out the resolution suitable for rendering at that capacity. This will avoid video re-buffering, and sudden drops in quality can also be avoided.

IV. PREDICTION OF VIDEO QUALITY DEGRADATION

From the observations of YouTube traffic pattern, we can say that by setting a threshold over the peak size and duration of a burst, we can predict when the video is going to switch from good quality to bad quality. Based on this we develop a mechanism to automatically identify the traffic bursts with the burst properties and accordingly trigger a notification when there is a possible drop in video quality.

For every resolution of video rendering, we define two thresholds - the peak size threshold ($\mathcal{T}_{PS}$) and the burst duration threshold ($\mathcal{T}_{BD}$). Whenever a data packet is received, we first identify whether the packet belong to the same burst or different burst. If the inter-arrival time between two consecutive packets are less than a threshold ($\mathcal{T}_{burst}$), we consider the packets to be in the same burst. For every such burst, we measure two parameters: burst age ($\mathcal{A}_b$) – the time difference between start of the burst and the reception time for current packet, and the estimated peak size ($\mathcal{P}_b$). Let δ be the amount of data (in bytes) received till now from the start of the present burst. Then, $\mathcal{P}_b$ is estimated as $(b * \mathcal{T}_{BD})/\mathcal{A}_b$. Based on the two thresholds ($\mathcal{T}_{PS}$, $\mathcal{T}_{BD}$) and two measured parameters ($\mathcal{A}_b$, $\mathcal{P}_b$), we define four streaming zones:

i) **Good Connectivity Streaming Zone (GCSZ):** $\mathcal{A}_b \leq \mathcal{T}_{BD}$ and $\mathcal{P}_b \geq \mathcal{T}_{PS}$ – this indicates bursts are short and sharp,
ii) **Bad Connectivity Streaming Zone (BCSZ):** $\mathcal{A}_b > \mathcal{T}_{BD}$ and $\mathcal{P}_b < \mathcal{T}_{PS}$ – this indicates bursts are small and wide,
iii) **Re-buffering Zone (RZ):** $\mathcal{A}_b > \mathcal{T}_{BD}$ and $\mathcal{P}_b \geq \mathcal{T}_{PS}$
iv) **Confusion Zone (CZ):** All other cases, the system can not say anything about the quality of the video.

Once these zones are identified, the system can send a trigger or early-alert to the video streaming server if the app is running at BCSZ. The video streaming server can take the preventive measures early enough, letting by reducing the video rendering resolution.

V. RESULTS AND CONCLUSION

Table I summarizes the results of our prediction mechanism. We observe that the traffic characteristics based network quality and video re-buffering prediction works with a good accuracy – we can detect the bad connectivity zones with 100% accuracy, although a few good connectivity zones sometime get predicted as a bad connectivity zone. This is due to the intermediate short term network quality fluctuation which our system fails to identify. Motivated by this initial good results, our future target is to develop an end-to-end video streaming application that exploit the traffic features for service quality management, and therefore can support better QoE for the end users.

REFERENCES

[1] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, 2011, pp. 133–144.
[2] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *Proceedings of the 10th ACM SIGCOMM IMC*, 2010, pp. 281–287.
[3] M. Z. Shafiq, J. Erman, L. Ji, A. X. Liu, J. Pang, and J. Wang, "Understanding the impact of network dynamics on mobile video user engagement," in *Proceedings of ACM SIGMETRICS*, 2014, pp. 367–379.