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. Such fluctuations lead to re-buffering and sudden drops in video quality, adversely affecting video watching experience. In this poster, we propose a light-weight method for early detection of network capacity degradation. 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 reveal that our proposed scheme works with very high accuracy.**

I. Introduction

The increasing penetration of smart-phones, tablets and other mobile devices is gradually changing the way end-users consume Internet content – particularly, multimedia content, 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], the prediction of end-user's quality of experience (QoE) often 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. Although metrics such as TCP goodput and current buffer size can be used to estimate connection quality, the former requires a fair amount of computation while the latter suffers from an extra level of indirection and is likely to be inaccurate. We propose an alternative light-weight method for early detection of user's QoE, which can be used to design a notification service in the client app; whenever QoE drops, the service can send notification to the content server, which then responds by taking steps for graceful degradation of video experience.

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.

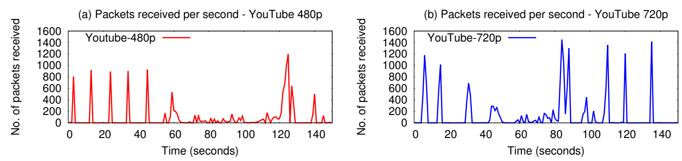

Fig. 1. YouTube Traffic Characteristics at Different Conditions

II. Data Collection

We have collected YouTube traffic traces from 4 different volunteers, through YouTube Android app installed in Moto-X second generation smart-phones. The data has been collected from both cellular and Wi-Fi networks at different traffic conditions with various indoor mobility patterns. 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. We have also manually tagged events such as re-buffering and auto-degradation of video resolution with timestamps. 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 "bad quality".

III. Analysis of YouTube Streaming Video Data

YouTube uses an adaptive buffering technique during video streaming, and therefore its traffic pattern shows a periodic bursty nature when the network condition is favourable, and the video is of good quality. Fig. 1 shows the traffic distribution

978-1-4673-9622-6/16/$31.00 ©2016 IEEE

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

of YouTube with respect to time, for two different resolutions of 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:

1) When the network quality is favourable 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, it affects the video rendering quality, and we see a gradual drop in the peak size of the bursts and the burst duration tends to become large (bursts widen).
4) During the re-buffering, we observe a wider burst duration, although with a high 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 favourable network, the peak size and the burst duration have 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, re-buffering occurs because the system fails to predict sudden drops in network capacity. As a consequence, the video is re-buffered and rendered at a very low resolution. If such drops can be predicted early, the app can send a notification to the video streaming server to take preventive measures by finding out the resolution suitable for rendering at that capacity. This will avoid video re-buffering, and sudden drops in quality.

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 belongs to the same burst or a different burst. If the inter-arrival time between two consecutive packets is 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 preventive measures sufficiently early, by reducing the video rendering resolution, and thereby ensuring graceful degradation of QoE.

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 good accuracy – we can detect the bad connectivity zones with 100% accuracy, although a few good connectivity zones sometimes get predicted as bad connectivity zones. This is due to the intermediate short term network quality fluctuation which our system fails to identify. Motivated by these initial good results, our future target is to develop an end-to-end video streaming application that exploits the traffic features for service quality management, and can therefore 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.