YouSlow: What Influences User Abandonment Behavior for Internet Video?

Hyunwoo Nam and Henning Schulzrinne
Columbia University, New York, NY

***Abstract*—With the proliferation of online video, measuring quality of experience (QoE) has become a crucial aspect for the analysis of today's over-the-top (OTT) video streaming. To monitor video QoE, we introduce YouSlow, a monitoring tool that can detect various playback events (e.g., start-up delay, rebufferings and bitrate changes) from video players while a video is being played. Using YouSlow, we have collected more than 1,400,000 YouTube views from more than 110 countries.**

We measured the impact of these playback events on video abandonment and found that about 10% of viewers abandoned YouTube videos when the pre-roll ads lasted for 15 seconds. We observe that more viewers abandoned YouTube videos when they suffered from the initial buffering than the rebufferings in the middle of a playback. Our regression analysis shows that bitrate changes do not affect video abandonment significantly and the abandonment rate can be estimated accurately using the rebuffering ratio and the number of rebufferings ($R^2 = 0.94$).

***Index Terms*—HTTP Video Streaming; Adaptive Bitrate (ABR) Streaming; Video Quality of Experience (QoE)**

I. Introduction

TODAY'S popular video streaming services such as Netflix, Hulu and YouTube stream video contents to viewers over HTTP or HTTPS. To provide smooth streaming, they use adaptive bitrate (ABR) streaming technologies such as Apple's HTTP Live Streaming (HLS) [1], Microsoft's Smooth Streaming (SS) [2], Adobe's HTTP Dynamic Streaming [3] and Dynamic Adaptive Streaming over HTTP (DASH) [4]. In ABR streaming, a video player dynamically adjusts video bitrates based on estimated network conditions, buffer occupancy and hardware specifications of viewers' devices, for example, distinguishing smartphones from desktops. Therefore, user-perceived video quality can vary depending on how appropriately the player selects the best available bitrate during a download. When the player requests a higher bitrate than what is actually available in the network, as an example, a viewer may experience frequent rebufferings, where the video is paused and then resumes playing repeatedly. It is also possible for the viewer to be stuck with a low bitrate during the entire playback if the network capacity is underestimated by the player. Hence, from over-the-top (OTT) video service provider's viewpoint, improving ABR heuristics is a key factor to enhancing video QoE.

To improve ABR streaming, it is important to analyze how changing ABR heuristics influences QoE. While traditional quality of service (QoS) based metrics, such as measuring TCP throughput, video packet delay and jitter, can be used to pinpoint network impairments, the metrics do not accurately reflect the viewer's watching experience. Thus, we believe that QoE monitoring systems should focus on application-layer events instead of transport-layer events. To achieve this, we suggest monitoring live playback events directly from within video players rather than network elements such as routers. As a proof of concept, we have developed YouSlow ("YouTube Too Slow!?"), a new QoE monitoring system for OTT streaming services. This lightweight web browser plug-in can monitor various playback events such as start-up delay, rebufferings and bitrate changes directly from within ABR players while viewers watch videos on the YouTube web site. So far, YouSlow has collected over 1,400,000 YouTube views from more than 1,000 viewers located in more than 110 countries.

In this paper, we evaluate various QoE metrics by analyzing video abandonment rates in YouTube. An abandonment occurs if a viewer closes the video during playback, either due to lack of interest or because they are annoyed by viewing interruptions such as long start-up delay, frequent rebufferings and bitrate changes. Below, we summarize our key findings and contributions:

- **Development of an analysis tool for video QoE:** YouSlow is designed to detect various playback events while a video is being played. Compared to prior approaches using survey-based metrics, YouSlow saves video researchers time and effort, particularly for large sample sizes. In addition, our QoE monitoring system allows viewers to track their viewing experiences such as average played bitrates and rebufferings in real time.
- **An analysis of video QoE in YouTube:** We observe that about 10% of viewers abandoned the videos when the pre-roll ads lasted for 15 seconds. We confirm that the initial buffering has more impact on the video abandonment than the rebufferings in the middle of a playback. Our analysis shows that viewers prefer constant bitrate to increasing bitrate during playback even if the abandonment rate is not significantly different. We show that tracking the rebuffering ratio during playback is useful to quantify abandonment rates for short videos. Our regression analysis using the rebuffering ratio and the number of rebufferings achieves an R-squared value of 0.94 in predicting the video abandonment rate in YouTube.

The remainder of this paper is organized as follows. In Section II, we briefly describe online video streaming and the principle of ABR streaming. Section III describes the overview of YouSlow and its implementation. Then, we present our analysis of YouTube in Section IV. Our QoE analysis report is described in Section V. We discuss the YouSlow platform in Section VI. Finally, we look at the related work and summarize our conclusions in Section VII and VIII, respectively.

II. Online video delivery background

Below, we briefly describe three popular streaming technologies; Real Time Messaging Protocol (RTMP) / Real Time Streaming Protocol (RTSP) streaming, *progressive download* and ABR streaming.

A. RTMP/RTSPhunk based delivery

The Adobe chunk-based delivery mechanism uses RTMP [5] and RTSP [6] for streaming video and audio data between Flash servers and Flash players. A special media server such as Flash Media Server [7] and Wowza [8] streams a series of video chunks and a Flash player consumes the content instantly without any local caching. The streaming server using dynamic RTMP [9] contains multiple bitrates for a single video file and allows the player to automatically change the bitrates during playback based on the network conditions. However, RTMP / RTSP streaming requires a special Flash-based media server.

B. Progressive download

In *progressive download*, a video is delivered by a regular web server using HTTP rather than a streaming server. This mechanism is easy to setup and cost-effective since it does not require any special streaming servers. When there is a video request, an HTTP web server pushes the video content as quickly as it can. The playback can start as soon as enough content has been downloaded and *fast forwarding* (skipping ahead) is only possible for the downloaded content. There is a security concern since the player caches the video content on the viewer's device. To prevent illegal copying of cached content, Digital Rights Management (DRM) can be used [10]. Moreover, *progressive download* provides no quality adjustment; no matter what download speed is experienced and what devices are used, the player downloads the same quality of the video file.

C. ABR streaming

Today's popular video streaming services such as YouTube, Netflix, HBO GO and BBC prefer ABR streaming technologies, having advantages of automatic quality switching and ease of delivery over HTTP. There are four popular ABR technologies: Apple's HLS [1], Microsoft's SS [2], Adobe's HDS [3] and 3GPP/MPEG DASH [4]. In ABR streaming, a video server contains multiple bitrates encoded for a single video object and each bitrate file is split into small segments. A segment size is measured in seconds (not bytes) and its length is typically between two and ten seconds. A manifest file contains the bitrate information such as the index of segments and their location. Before playback, an ABR player downloads the manifest file and it dynamically adapts the bitrate based on CPU availability and network conditions while a video is being played. In ABR streaming, viewers need to install plug-ins for on different types of devices (mobile and desktop), OSs and ABR streaming technologies. For example, Adobe's HDS requires Flash plug-ins. For Microsoft's SS, viewers need to install the Silverlight extensions in their web browsers. Apple's HLS supports all Apple's devices but may not properly work on old Android devices. Because of this, today's OTT streaming services such as YouTube and Netflix prefer HTML5 video that most web browsers (Safari 3+, Internet Explorer 9+, Firefox 3.5+, Chrome 3+ and Opera 10.5+) support. HTML5 enables MPEG DASH native playback using Media Source Extensions (MSE) that allows JavaScript to deliver media streams for playback within web browsers [11].

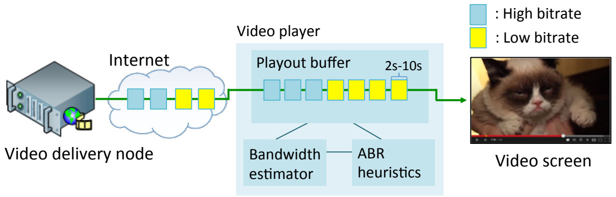
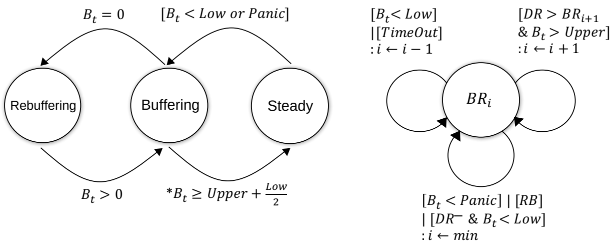


Fig. 1: How does ABR streaming work?

Figure 1 depicts an ABR streaming platform. To better comprehend ABR streaming, we need to first understand how the video encoding for ABR streaming differs from the encoding for *progressive download*. To decode a video properly, a player needs to download an I-frame (intra frame), also known as a key frame, while a video is being played. In *progressive download*, an MPEG I-frame is inserted periodically (e.g., every ten seconds) into a single video file. In ABR streaming, a source video is encoded into multiple different files, each at different bitrates, and each such file is divided into a series of small segments. Each segment contains at least one key frame, preferably at the beginning of the segment. Depending on encoding tools, a single segment may have multiple key frames. For example, according to the technical note for Apple's HLS [12], a segment size is ten seconds and the key frame interval is three seconds.

As noted, a segment size is generally between two and ten seconds long. Smaller segment sizes lead to decreased encoding efficiency in terms of GOP (group-of-pictures) frame size. Because of the higher number of segments, more I-frames are needed in the final encoding. On the other hand, longer segment sizes may cause frequent rebufferings under unreliable network conditions. For example, let's suppose that an ABR player is downloading a segment and the network is congested. The segment size is ten seconds and uses a single key frame. We note that an ABR player can switch bitrate only at an I frame. In this case, the player is unable to select lower bitrates until the requested segment has been downloaded in full. If the playout buffer is nearly empty, this may cause frequent rebufferings in the middle of the download. To prevent this, the player may use a timer; when the timer expires, it abandons the downloading segment and requests a low-quality segment. But this may cause frequent bitrate changes if the timer length is too short. If the segment size is



* It depends on ABR configuration

Fig. 2: Finite state machine (FSM) of state change and bitrate switching behavior of Microsoft's SS players

shorter (e.g., two or five seconds instead of ten seconds), the player can handle this situation better by switching to lower bitrates more quickly.

In ABR streaming, a player uses a set of heuristics to find the best available bitrate during playback. Based on our own analysis and technical overview of ABR streaming [13], [14], the following inputs are generally considered in picking the bitrate:

- real-time available network bandwidth and amount of video remaining in the playout buffer during playback;
- screen resolution and video rendering capabilities of viewers' devices;
- frame rate and viewers' interactive actions (e.g., resizing the browser window) during playback.

A player may experience frequent frame drops when a system is running multitasking that requires significant RAM and CPU usage. When a large number of frames is dropped, the player flushes its buffer and re-downloads the discarded segments at lower playback rates to provide a good video quality.

While a video is being played, the state of video player can be Buffering, Steady or Rebuffering. During Buffering state, the ABR player aggressively downloads video content into its playout buffer. The player requests the next segment right after it completely downloads the current segment (back-to-back HTTP requests) so that the buffer can be filled as quickly as possible. When the playout buffer is above a configured threshold, the player goes into Steady state. Instead of increasing the playout buffer level by downloading the segments back-to-back, the player in Steady state tries to keep the buffer full. In order to avoid buffer overflow, it requests one segment for every segment duration. When the playout buffer is running low, the state will switch to Buffering again. Rebuffering is referred as buffer stalling or video buffering. It occurs when there is no video content available in the playout buffer during playback.

We examine the source code of the Silverlight extension, an ABR player for Microsoft's SS [15], and summarize the ABR player's state change and bitrate switching behaviors in Figure 2. The parameters are described as follows:

- B_t represents how much video content is currently left in the playout buffer (in seconds).
- BR_i represents the video bitrate (in kb/s) selected by a player during playback, where $BR_{min} \leq BR_i \leq BR_{max}$. BR_{min} indicates the minimum bitrate and BR_{max} presents the maximum bitrate among the available bitrates of the video.
- Panic, Low and Upper: An ABR player takes into account three pre-defined thresholds when it changes state (Buffering or Steady) and picks the best available bitrate during playback. For example, when a buffer level becomes lower than the Low threshold, the bitrate will be downgraded by one step (BR_i ➜ BR_{i-1}). When the buffer level is lower than the Panic threshold, it directly drops down to the lowest bitrate (BR_i ➜ BR_{min}). When the buffer level is higher than the Upper threshold and the measured network throughput is larger than the next bitrate, then the current bitrate will be increased by one step (BR_i ➜ BR_{i+1}). The bitrate can be increased or decreased by multiple steps at a time (e.g., two or three steps) when the available bandwidth is changing rapidly. Before the player attempts to increase the bitrate, it waits for a certain amount of time such as three or five seconds to prevent frequent bitrate changes. All these settings depend on the ABR configuration.
- TimeOut: The timer is set to estimate network conditions. It activates when the elapsed time for downloading a requested segment is longer than the expected time. In such case, the bitrate is decreased for the next request.
- DR denotes the current downloading data rate measured by a bandwidth estimator in an ABR player. DR^{-} indicates that the available bandwidth in the network is decreasing. The time period required for the estimator to analyze the network conditions depends on the ABR configuration. For example, the estimator measures the average download throughput over the most recent three or five segments.

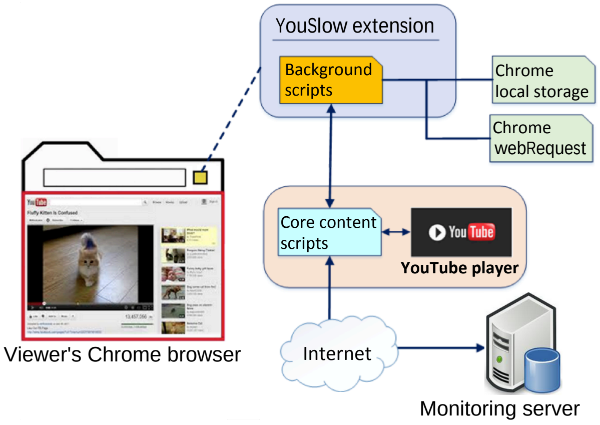


Fig. 3: YouSlow Chrome plug-in for YouTube analysis

III. YouSlow overview

YouSlow can monitor various playback events directly from within an ABR player for an analysis of video QoE. Currently, YouSlow only supports YouTube, but other players' JavaScript APIs such as Vimeo [16] could be easily added to YouSlow.

A. Implementation

YouSlow supports three different platforms: The Chrome web browsers, iOS and Android. We have recently released the beta versions of iOS and Android applications on the YouSlow web site[1]. For desktops and laptops, we created a lightweight Chrome plug-in, also known as a Chrome extension [17]. We distribute the YouSlow applications via the Chrome web store[2]. The source code is available in GitHub[3].

Figure 3 shows the architecture of the Chrome plug-in for YouTube analysis. YouSlow runs in the background of the Chrome browser, and injects our QoE monitoring scripts into the web page whenever a viewer watches a video on the YouTube web site, www.youtube.com. The YouSlow scripts contain YouTube player's iframe and JavaScript APIs [18] to access and monitor playback events of HTML5 and Flash video players. When a viewer ends a video session, the extension automatically reports the measurements to our monitoring server[1]. The collected data is analyzed and then marked on Google maps. For privacy reasons, the extension does not collect any information regarding the viewer's YouTube account, video URLs or video titles. Through our monitoring system[1], viewers can monitor various metrics about their YouTube watching experiences, such as how often they experience rebufferings and what video bitrates they typically watch. Using this information, they may compare the performance of their own ISPs with other local ISPs. Additionally, YouSlow outputs can be useful to video service providers to improve their ABR streaming services. For example, they can monitor and compare the rebuffering statistics every time there is a change in their ABR heuristics.

B. What factors can YouSlow measure?

YouSlow can measure the following factors during video playback:

- **Start-up latency:** YouSlow measures the start-up delay, the time from the instant a play button is clicked to when the player actually starts to play the main video. Thus, this includes initial buffering and pre-roll ads in YouTube.
- **Pre-roll ads:** YouSlow detects if the viewer experiences a pre-roll ad by inspecting HTTP GET URLs. YouTube uses different URL parameters when an ABR player downloads ads and the main videos.
- **Rebuffering:** YouSlow monitors the duration of rebuffering and how often it occurs during playback.
- **Bitrate change:** YouSlow measures how much an ABR player increases or decreases the bitrate every time it switches bitrates during playback.
- **Video loaded fraction:** YouSlow monitors the percentage of the video that the player shows as buffered. We calculate the fraction by dividing the total amount of downloaded video data by the full size of the video. For example, if the player downloads 10 MB from a 100 MB video, the fraction will be 0.1.
- **Location:** An IP geo-location database[4] pinpoints the approximate location (city, state, and country) of the playback event, and find the domain names of local ISPs that the viewers are connected to.

TABLE I: YouSlow dataset

Country [1-5]	Views	Country [6-10]	Views
United States	461,557	South Korea	55,559
United Kingdom	100,748	Indonesia	46,218
India	96,801	Canada	43,864
Malaysia	72,477	Philippines	31,238
Germany	57,876	Italy	29,998

(a) Top 10 countries

U.S. ISP [1-4]	Views	U.S. ISP [5-8]	Views
Comcast	92,660	Time Warner Cable	41,873
AT&T	91,231	Qwest Communications	18,708
Verizon	61,141	Century Link	13,396
Charter Communications	49,401	Frontier Communications	13,145

(b) Top 8 U.S. ISPs

IV. YouTube measurements

We analyzed 1,471,958 YouTube views collected between February 2015 and July 2016 from more than 1,000 viewers in 117 countries. We note that the dataset only includes the video sessions where the viewers watched videos through the YouTube web site using the Chrome browser on desktops or laptops. Table Ia shows the top ten countries along with the total number of reported views. We also compare and analyze the measurements for different U.S. ISPs (Table Ib).

A. Start-up delay

We measure the elapsed time from when a play button is clicked to when the main video starts. There are two factors that contribute to start-up delay: initial buffering and pre-roll ads. For initial buffering, an ABR player typically downloads a few segments (two or three) before it starts to begin playback. The required number of segments depends on ABR configuration. For example, the player may store a larger number of segments (five or ten) to avoid future bandwidth fluctuations at the beginning of playback [19]. Secondly, an ABR player does not play the selected video until viewers have watched the pre-roll video ad. YouTube's advertising policies [20] describes two types of video ads: *skippable* and *non-skippable*. *Skippable* video ads allow viewers to skip the ad after five seconds. *Non-skippable* video ads must be watched to view the main video and they are usually 15-20 seconds long [21]. Both types of ad can appear before, during or after the main video. YouSlow is not able to distinguish if the ads are *skippable* or *non-skippable*. The ad length recommended by YouTube is less than 3 minutes. The post-roll ads are typically not effective because most viewers close videos once they have watched the main content. The viewers who use an ad-block extension [22] may be able to watch the entire video without ads. We observe that the player uses different URL parameters for downloading the video ads and the main video. To distinguish them, we use the Chrome webRequest API [23]. Currently, YouSlow focuses on the analysis of pre-roll ads in YouTube.

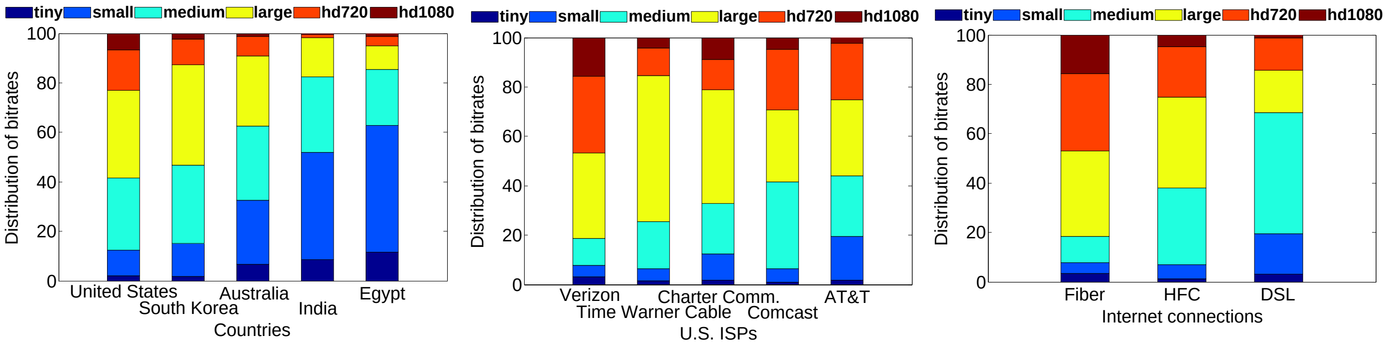
Figure 5a presents the cumulative probability of the initial buffering duration. Figure 5b shows the cumulative probability

[1] YouSlow - https://dyswis.cs.columbia.edu/youslow/

[2] Chrome web store - http://goo.gl/AIOED3

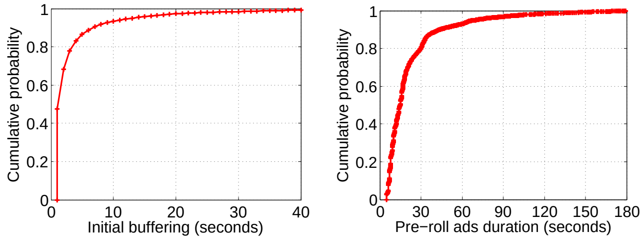
[3] YouSlow GitHub - https://github.com/leftdal/youslow

[4] MaxMind GeoIP database - http://dev.maxmind.com/



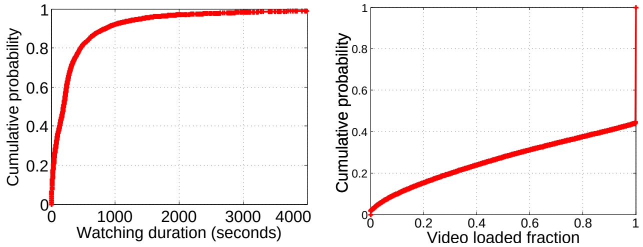
(a) Countries **(b)** U.S. ISPs **(c)** Different types of Internet connections

Fig. 4: Comparison of YouTube played bitrates



(a) Initial buffering **(b)** Pre-roll ads length

Fig. 5: Start-up delay caused by initial buffering and pre-roll ad



(a) YouTube watching duration **(b)** Video loaded fraction

Fig. 6: Video watching duration and video loaded fraction

of how long the viewers watched the pre-roll ads before the main content. Compared to the pre-roll ads, the initial buffering has a relatively shorter duration.

B. Video watching duration

We measure how long a viewer stays in each video session. The watching duration also includes rebuffering and start-up latency. Based on the experimental results in Figure 6a, we observe that the average of watching duration is 6:36 minutes per video session and the median is 2:39 minutes.

C. Video loaded fraction

We measure video engagement by monitoring the video loaded fraction for the main video. According to Figure 6b, more than 40% of viewers closed YouTube videos in the middle of the playback. They may have lost interest in the videos or suffered from unexpected viewing interruptions such as video ads, rebufferings and bitrate changes.

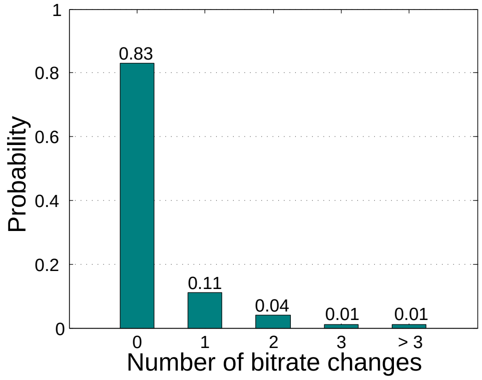


Fig. 7: Probability of number of bitrate changes

D. Bitrate changes

We observe that most video sessions ($>99\%$) experience fewer than four bitrate changes during playback. Figure 7 shows the probability mass function (PMF) of bitrate changes in the dataset. 83% of video sessions in YouTube did not change bitrates during the entire playback.

E. Played bitrates

According to YouTube's encoding policies [24], YouTube streams eight different bitrates: `highres`, `hd1440`, `hd1080`, `hd720`, `large`, `medium`, `small` and `tiny`. We describe each bitrate setting in Table II, and measured the distribution of played bitrates in Table III. These measurements indicate that most viewers on desktops or laptops watched YouTube videos with `large` (33.1%) or `medium` (23.7%) bitrates. We also observed a few `hd1440` and `highres` videos, but the probability is much smaller ($<0.1\%$).

In Figure 4a, we compare the distributions of played bitrates across countries. For example, viewers in the United States and South Korea experienced higher bitrates in comparison to the ones in India and Egypt. Figure 4b shows the distributions of played bitrates for different ISPs in United States. For more details, we compare the distributions depending on different types of Internet connections such as fiber to the home (FTTH), hybrid fiber-coaxial (HFC) and digital subscriber line (DSL). We collected 7,074 samples in total for FTTH

TABLE II: YouTube bitrate setting

Type	Video bitrate	Resolution
highres	35 - 45 Mb/s	3840×2160
hd1440	10 Mb/s	2560×1440
hd1080	8 Mb/s	1920×1080
hd720	5 Mb/s	1280×720
large	2.5 Mb/s	854×480
medium	1 Mb/s	640×360
small	400 kb/s	426×240
tiny	80 kb/s	256×144

TABLE III: YouTube played bitrates (%)

hd1080	hd720	large	medium	small	tiny
6.8%	18.2%	33.1%	23.7%	13.3%	4.9%

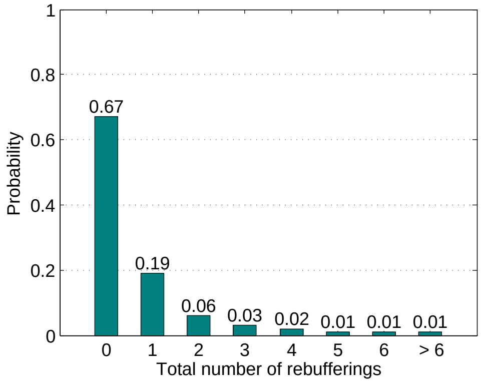


Fig. 8: Probability of total number of rebufferings

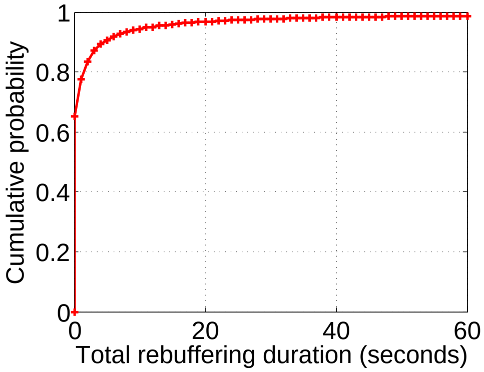


Fig. 9: Cumulative probability of total rebuffering duration

from Verizon's *FiOS* Internet service, and 6,618 samples for HFC from Time Warner Cable, Charter Communications, Cox Communications, Comcast and AT&T's *U-verse* (formerly *Project Lightspeed*). For DSL, we obtained 2,384 samples from Verizon (*non-FiOS*), AT&T (*non-U-verse*) and Qwest Communications. YouSlow can distinguish these by comparing the hostnames of the Internet service providers of the viewers using the IP geo-location database[4]. For example, Verizon hosts consistent domain names (e.g., x.x.fios.verizon.net) for their *FiOS* users. Through the measurements, we observe that the viewers using fiber watched more videos at HD bitrates (36.8%) than the ones using HFC (25.3%) or DSL (14.4%).

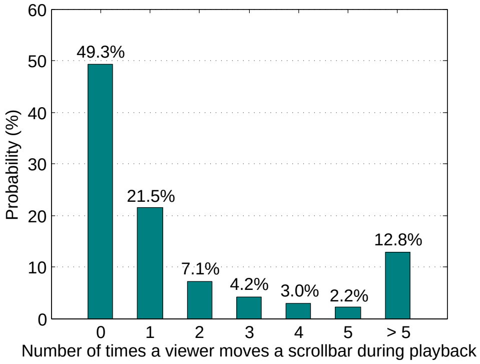


Fig. 10: Probability of total number of times a viewer moves a scrollbar during playback

F. Rebufferings

Figure 8 shows the PMF graph of total number of rebufferings. In the dataset, we find that more than 99% of video sessions suffered from fewer than six rebufferings and 67% of viewers experienced no rebufferings during the entire playback. Figure 9 shows the cumulative probability of total rebuffering duration per video session. Our experimental results show that only 10% of rebufferings exceeds 10 seconds in total.

G. AdBlock extension

The YouSlow Chrome extension (version 1.2.8) is able to detect if a viewer uses an adBlock extension [22] on the Chrome web browser while a video is being played. Using the extension, a viewer may watch the YouTube videos without experiencing ads during entire playback. The extension investigates the URLs and HTML elements on the web page and blocks them if they are known as advertisement. We analyzed a total of 124,744 video sessions and found that 84,698 (67.9%) videos were watched on the Chrome web browsers using the adBlock extension.

H. Moving a scrollbar of YouTube player during playback

The YouSlow Chrome extension (version 1.2.8) is able to detect if a viewer is moving the scrollbar of YouTube player forwards or backwards during playback. We analyzed a total of 125,277 video sessions. Our analysis in Figure 10 shows that about 50% of total viewers moved the scrollbar at least one time during the entire playback.

V. Video QoE analysis in YouTube

In this section, we describe our analysis of video QoE based on YouSlow measurements. We are trying to answer the following questions:

- How do start-up delay, rebufferings and bitrate changes affect viewing interruption?
- What metrics can we use to analyze the impact of the above playback events on video QoE?

A. QoS and QoE methods for an analysis of video QoE

1) QoS methods: Several researchers [25]–[27] have used QoS-based metrics such as monitoring throughput, goodput, packet delay and jitter from intermediate nodes such as routers between viewers' devices and video servers, to analyze the performance of video streaming. This approach typically focuses on finding network impairments, but there are challenges to estimating QoE for buffered video streaming. As an example, periods of low TCP throughput do not always interrupt a viewer's watching experience if an ABR player has downloaded enough data into the playout buffer. The QoS-based metrics are unable to detect the impact of low-throughput period since they cannot accurately track the playout buffer level from within the network routers.

Today's popular video streaming services such as YouTube and Netflix provide a video quality report to viewers [28], [29]. They simply measure the download speed from their own video content servers or video applications and rate the video streaming quality for different ISPs and geographical location of viewers. However, the output does not provide any QoE metrics to viewers, such as how often they experience bitrate changes and rebufferings.

2) QoE methods: In terms of the QoE definitions by ITU-T [30], it is the overall acceptability of an application or service, as perceived subjectively by the end-user. For video QoE, it is a perceptual measure that reflects viewers satisfaction with their video streaming experience. The common approach is a subjective method, hiring a group of people, having them watch short video clips, and scoring their viewing experiences under the laboratory or the crowd-sourcing environments [31]–[33]. However, such survey-based metrics are typically costly and time-consuming. In addition, it is difficult to automate and control the testing environments during the evaluation. To avoid high cost of subjective methods, objective methods are developed to estimate QoE of viewers [34]–[36]. This method focuses on building a statistical model based on various input QoS parameters measured from network or service layers such as packet loss, throughput, bitrate and video frame loss. However, it is hard to develop such QoE prediction models. For example, any modification made to current models such as adding or removing input parameters may require new tests to create new models. For evaluation, it also requires the survey-based methods. Due to the above limitations, today's many video researchers focus on the data-driven analysis [37]–[40]. The common approach is from quality of "*experience*" such as happiness or satisfaction to quality of "*engagement*" such as video abandonment and failure [38]. The user-engagement metrics can be easily quantified and measured since it does not require direct user-involvement such as the mean opinion score (MOS) metrics that most subjective methods use.

3) Our analysis methodology and metrics: YouSlow is able to detect the video abandonment event directly from within the video player. Since the video rendering quality and level of interest are independent, we believe that our results are relatively insensitive to changes in how we define QoE-driven abandonment. We compute QoE abandonment ratio by dividing the number of sessions abandoned due to viewing interruptions such as rebufferings and bitrate changes by the total number of video sessions. For instance, we calculate an abandonment rate depending on how long viewers suffer from pre-roll ads and rebufferings and analyze the impact of bitrate changes on abandonment by comparing with constant bitrate videos. With a large number of samples, we believe that monitoring abandonment rates gives us practical and reliable outputs to analyze viewing interruptions in online video streaming.

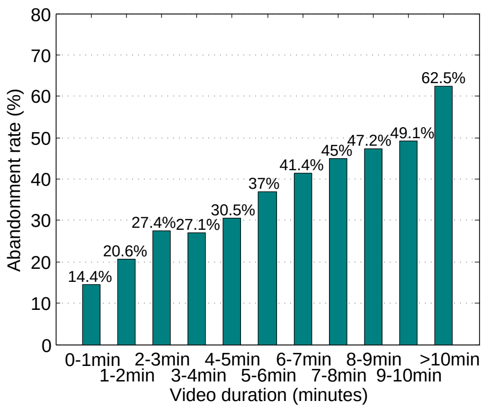


Fig. 11: Abandonment rate (%) for unimpaired videos with different video lengths

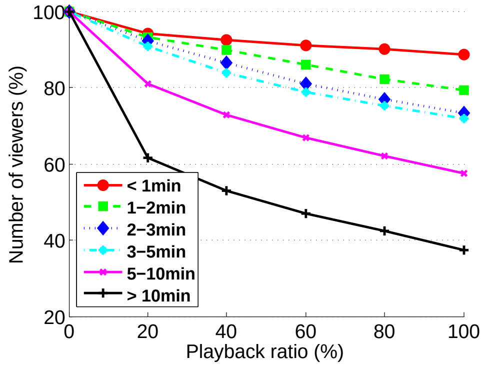


Fig. 12: Number of viewers (%) for unimpaired videos with different video lengths as playback ratio increases

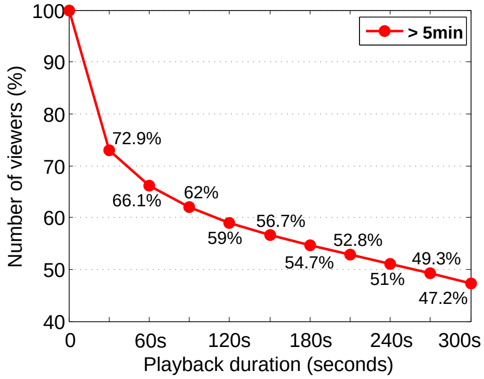


Fig. 13: Number of viewers (%) for unimpaired videos (length >5 minutes) as playback duration increases

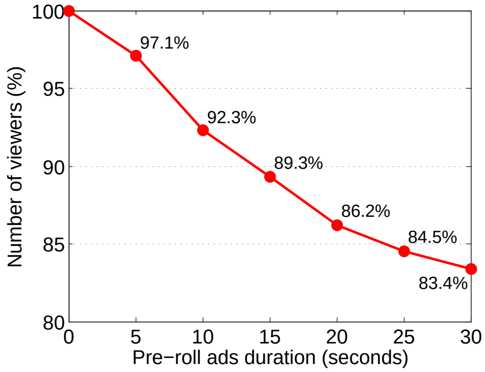


Fig. 14: Number of viewers (%) during pre-roll ads as ad duration increases

B. QoE analysis report

Below, we summarize our QoE analysis based on YouSlow measurements in YouTube. The video samples are grouped and analyzed using the following notations:

- **Unimpaired videos:** The viewers watched the videos without any viewing interruptions such as video ads, long initial buffering (> 1 second), rebufferings and bitrate changes.
- **Ad-free videos:** The viewers watched the videos without experiencing pre-roll ads before the main content.
- **Rebuffered videos:** The viewers suffered from rebufferings int the middle of playback.
- **Initial buffered videos:** The viewers experienced a long initial buffering (> 1 second) at the beginning of playback.

1) Video length: As a baseline analysis, we analyzed the abandonment rate depending on the video length. Concentrating on the impact of video length, we analyzed the unimpaired video sessions only. Figure 11 shows our experimental results. We find that an abandonment rate rises as video length increases. Most viewers decide whether or not they want to watch the video at the beginning of playback [41]. Figure 12 shows how many viewers stayed in the video sessions as the playback ratio increased for different video lengths. The playback ratio shows the ratio of content that has played in a video. For the videos that are longer than 10 minutes, for instance, we find that only 60% of viewers stayed in the video sessions when the playback ratio is 0.2. In addition, we analyzed the number of viewers as the playback duration (in seconds) increased for the videos that are longer than 5 minutes. As shown in Figure 13, about 44% of viewers abandoned the videos during the initial 60 seconds and then they started to abandon at slower rates.

2) Pre-roll ads: We have recently added a pre-roll ad analysis function to our Chrome extension (version 1.2.7). We analyzed a total of 11,038 video sessions where the viewers experienced the pre-roll ads. Among them, 2,635 videos were abandoned during the ads. Based on this analysis, the abandonment ratio for pre-roll ads is about 23.9%. Figure 14 shows how many the viewers stayed during the pre-roll ads as the ad length increased. We conjecture that most of the abandonment in Figure 14 took place during the *non-skippable* ads. Otherwise, the viewer may skip the ads instead of abandoning the videos. *Non-skippable* ads are usually 15-20 seconds long. Our experimental results show that about 10% of viewers abandoned the videos when the ads lasted for 15 seconds in YouTube.

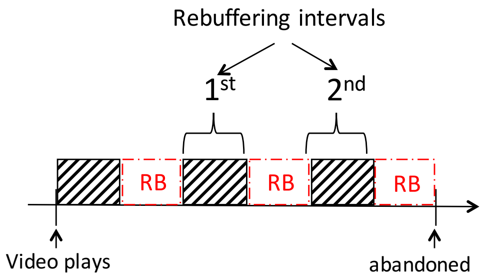


Fig. 15: Two rebuffering (RB) intervals with three rebufferings

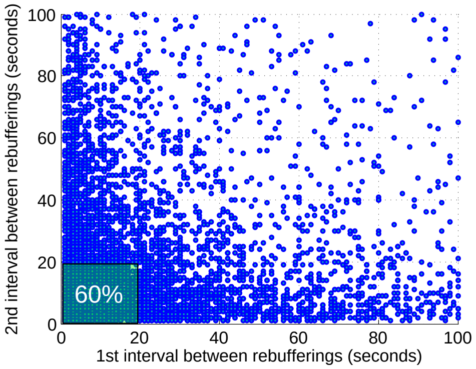


Fig. 16: Plotting abandonments for rebuffered videos with three rebufferings

3) Rebuffering: Most recent studies on video QoE [25], [42]–[45] agree that rebufferings should be avoided if at all possible in order to enhance video QoE. In addition, they show that QoE of viewers can vary depending on the rebuffering pattern, i.e., how many or how often rebufferings appear during playback. We try to understand how viewers react to such different rebuffering patterns in YouTube, along with abandonment rates. As a baseline analysis, we extract the video sessions from the dataset where the total number of rebufferings is three (Figure 15), and plot the abandonments based on the rebuffering intervals (Figure 16).

In Figure 16, we observe 60% of abandonments when the rebufferings intervals are less than 20 seconds. We frequently observe such short rebuffering intervals when an ABR player requests a higher bitrate than what a network can handle. In this case, the video play has to be paused until the player stores at least one segment in the buffer, which can cause a series of short-term rebufferings. Furthermore, we observe that an abandonment pattern varies depending on rebuffering intervals. For instance, let's suppose that we have a certain range of first rebuffering interval between 0 and 20 seconds in Figure 16. Depending on the second interval, we clearly see that the distribution of abandonments varies. The question is, how do we normalize the impact of rebuffering intervals and correlate the results with QoE assessments such as MOS? If we take into account a higher number of rebufferings or additional

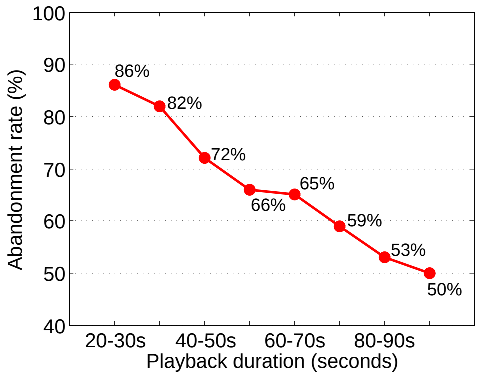


Fig. 17: Abandonment rates (%) for ad-free, non-initial buffered and rebuffered videos - the same rebuffering duration but with different playback durations

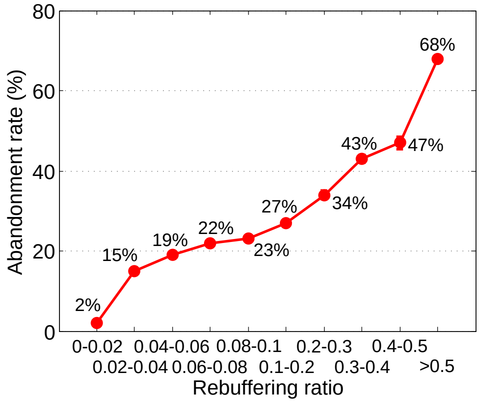


Fig. 18: The impact of rebuffering on abandonment rates (%) for ad-free, non-initial buffered and rebuffered videos

TABLE IV: Number of samples for rebuffering ratio analysis

Rebuffering ratio	Number of samples
0 - 0.02	143,799
0.02 - 0.04	19,584
0.04 - 0.06	8,998
0.06 - 0.08	5,210
0.08 - 0.1	3,403
0.1 - 0.2	6,776
0.2 - 0.3	2,521
0.3 - 0.4	1,462
0.4 - 0.5	984
>0.5	6,649

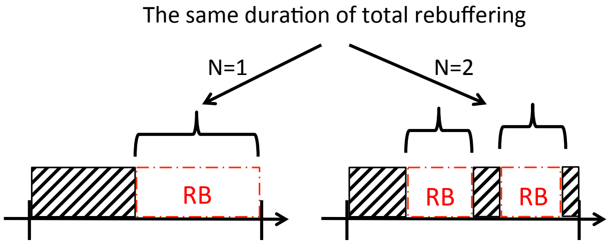


Fig. 19: The same total rebuffering (RB) duration with different number of rebufferings

factors such as rebuffering duration and total playback length, QoE modeling will be much more complicated. To avoid such complexity, we consider a simpler metric below and analyze how will the metric predict the abandonment rate.

Rebuffering ratio: The total rebuffering duration is not sufficient for modeling QoE metrics since it does not take into account the total duration of playback. For example, viewers may experience watching interruptions differently depending on total playback duration, even if the video session has the same duration of total rebuffering. As an example, we calculate the abandonment rates for the video sessions where the total rebuffering duration is between 10 and 15 seconds but they have different total playback durations (20 through 100 seconds). Figure 17 shows our experimental results. We clearly see that the abandonment rate decreases when the playback duration increases.

$$\text{RB ratio} = \frac{\text{Rebuffering duration (second)}}{\text{Total playback duration (second)}} \qquad (1)$$

To reflect this, we analyze the impact of rebufferings on abandonment rates using the rebuffering (RB) ratio in Equation 1. The ratio is defined as the fraction of time when a viewer experiences rebufferings while watching a video. As an example, rebufferings occur for ten seconds while a viewer stays in the video session for 100 seconds. In this case, the rebuffering ratio will be $10/100 = 0.1$. Depending on the rebuffering ratio, we calculate the abandonment rate. To avoid the video abandonment due to the lack of interest at the beginning of playback, we analyze the video sessions where the viewers watched the videos for at least 60 seconds. Figure 18 shows average abandonment rate as rebuffering ratio increases. We plot the results with the error bars based on the standard error of the mean (SEM), but the error values are less than 0.2% for each rebuffering ratio. We subtracted the unimpaired rate (RB ratio = 0) from the impaired rates. The abscissa indicates a range of rebuffering ratio ($x-y$ represents $x < \text{ratio} \leq y$). Table IV shows the number of samples for each rebuffering ratio. The results tell us that more viewers abandoned the videos as the rebuffering ratio increased.

We note that the rebuffering ratio does not take the number of rebufferings into account. As shown in Figure 19, for instance, it is possible that the number of rebufferings can vary although the total rebuffering duration is the same. This can affect video QoE differently. To prove it, we compare the impact of a single rebuffering event and multiple rebufferings by comparing the abandonment rates along with rebuffering ratio. In Figure 20, we clearly observe that multiple rebufferings cause higher abandonment rates than a single rebuffering event.

Rebuffering early vs. later: We analyzed the impact of rebuffering start time on video abandonment. We collected the number of video sessions with abandonment due to rebufferings or initial buffering and counted the number of abandonment depending on the rebuffering start times. Figure 21 shows our experimental results. During the experiment, we analyze the video sessions with a single rebuffering event to avoid the impact of multiple rebufferings. We observe that the viewers were more likely to close the videos when they experience the initial buffering at the beginning of a playback. The initial

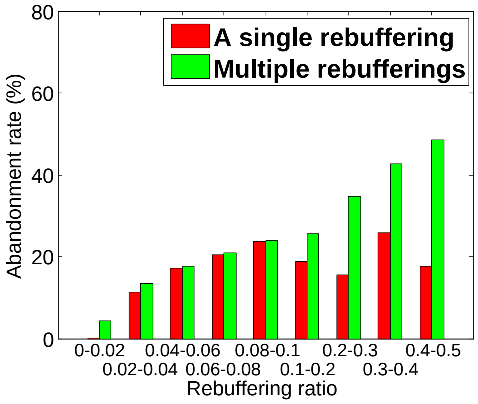


Fig. 20: Comparison of abandonment rates between a single rebuffering event and multiple rebufferings for ad-free, non-initial buffered and rebuffered videos

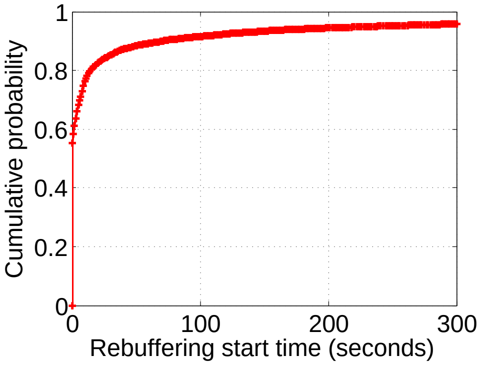


Fig. 21: Cumulative probability of number of abandonments for ad-free, initial buffered and rebuffered videos with different rebuffering start times

buffering is related to the ABR heuristics of selecting bitrates and the network conditions when the video starts. For example, an ABR player may request high bitrate segments when it starts downloading a video. If the high bandwidth is available, it can play such high quality of video instantly, increasing the video QoE. However, it may experience a long initial buffering at the beginning of a playback if the network is congested.

4) Bitrate switching: Some papers [46]–[50] investigate the impact of bitrate changes on video QoE. They claim that providing a bitrate as high as possible does not necessarily lead to the highest QoE [47]. They agree that it is difficult to create a metric that takes into account of all the bitrate switching events, such as the number of bitrate changes, their amplitude (i.e., by how much bitrate increases or decreases) and the duration of each bitrate. Below, we try to find a simple metric that can properly reflect and quantify the impact of bitrate changes on abandonment rates in YouTube.

Bitrate change ratio: To find the impact of bitrate changes on abandonment rates, we use the following equation:

$$\text{BR change ratio} = \frac{\sum_{\text{i=1}}^{\text{Num. of BR changes}} |\log(\text{BR}_\text{i}/\text{BR}_{\text{i}-1})|}{\text{Num. of BR changes}} \quad (2)$$

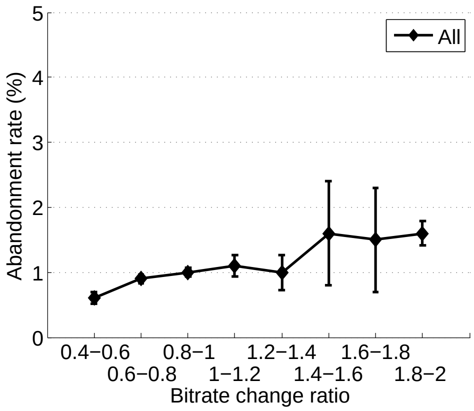


Fig. 22: The impact of bitrate change on abandonment rates (%) for ad-free, non-initial buffered and non-rebuffered videos

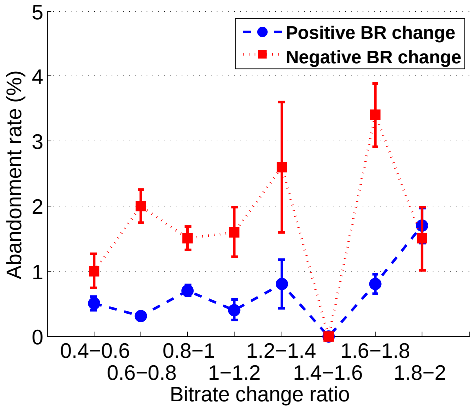


Fig. 23: The impact of positive and negative bitrate changes on abandonment rates (%) for ad-free, non-initial buffered and non-rebuffered videos

BR_i and BR_{i-1} denote the newly selected bitrate and the previous bitrate (in kb/s), respectively. Using the above equation, we calculate the abandonment rates. To remove the influence of other factors such as rebufferings and ads, we first collect the video sessions with bitrate changes only. To avoid counting the cases where a video is closed due to lack of interest, we only considered the videos as abandoned when they were watched at least 60 seconds and closed within five seconds after the bitrate was changed in the middle of a playback. Figure 22 shows our experimental results. We plot the results with the error bars based on the standard error of the mean (SEM). Table V shows the number of samples for each bitrate change ratio. The analysis indicates that the viewers were more likely to close the videos when the bitrate change ratio increased (Figure 22). However, the abandonment rate is not significantly different ($<2\%$).

The above result leads to the following question: does switching to a higher bitrate during playback also increase abandonment rate? To figure this out, we analyzed the video sessions with positive or negative bitrate changes separately. In Figure 23, positive bitrate changes present the views where there was only bitrate increase during video playback (e.g., $BR_i - BR_{i-1} > 0$) and negative bitrate changes present the

TABLE V: Number of samples for bitrate change ratio analysis

Bitrate change ratio	All	Positive	Negative
0.4 - 0.6	6,684	3,883	1,425
0.6 - 0.8	16,200	9,596	2,862
0.8 - 1	20,780	10,561	4,546
1 - 1.2	3,359	1,522	963
1.2 - 1.4	1,478	515	45
1.4 - 1.6	198	14	6
1.6 - 1.8	7,009	3,484	1,284
1.8 - 2	3,690	1,918	1,440

TABLE VI: The impact of a single bitrate (BR) change on video loaded fraction for ad-free, non-initial buffered and non-rebuffered videos with different starting bitrates and total playback lengths

`tiny` or `small`	0 - 60 s	60 - 120 s	120 - 180 s	$>$ 180 s
No BR changes	0.41	0.78	0.9	0.94
Positive BR change	0.38	0.69	0.85	0.9
Negative BR change	0.24	0.63	0.83	0.89
`medium` or `large`	**0 - 60 s**	**60 - 120 s**	**120 - 180 s**	**$>$ 180 s**
No BR changes	0.44	0.83	0.92	0.95
Positive BR change	0.43	0.77	0.87	0.92
Negative BR change	0.33	0.7	0.83	0.9
`hd` or `highres`	**0 - 60 s**	**60 - 120 s**	**120 - 180 s**	**$>$ 180 s**
No BR changes	0.48	0.8	0.91	0.94
Positive BR change	0.43	0.74	0.86	0.92
Negative BR change	0.4	0.7	0.83	0.89

TABLE VII: The impact of bitrate (BR) change on video loaded fraction for ad-free, non-initial buffered and non-rebuffered videos with constant bitrate or multiple bitrate changes

Avg. played bitrate (kb/s)	Constant BR	Multiple BR changes
350 - 450 kb/s	0.9	0.81
900 - 1,100 kb/s	0.92	0.83
2,400 - 2,600 kb/s	0.93	0.82

views where there was only bitrate decrease during playback (e.g., BR_i - $BR_{i-1} < 0$). We clearly observe that decreasing bitrate causes higher abandonment rate than increasing bitrate during playback. Note that we have collected a small number of samples for the bitrate change ratio between 1.4 and 1.6 (Table V) and all the viewers for the range completely watched the videos until the end.

The impact of bitrate changes on video loaded fraction: We compare the impact of positive and negative bitrate changes on the video loaded fraction. To remove the impact of multiple bitrate changes, we analyzed the video sessions that experienced a single bitrate change only. We split the dataset into three groups depending on different starting bitrates such as `tiny`/`small`, `medium`/`large` and `hd`/`highres`. Table VI shows the average of video loaded fraction based on the total playback length. We observe low video loaded fraction for short playback duration (0 - 60 seconds). This typically happens when the videos are not what the viewers expected in YouTube. In this cases, the videos are easily abandoned at the beginning of playback, which results in the low loaded fraction. The viewers typically stayed longer in the video sessions when they watched high (`medium` or above) bitrates at the beginning of a playback. In addition, the viewers were likely to abandon videos early when the bitrates decreased, but the video loaded fraction is not significantly different. Interestingly, we also observe that more viewers abandoned the videos early even when the players increased the bitrates regardless of starting bitrate and playback length.

We analyze the impact of bitrate changes depending on the average played bitrate. For instance, let's suppose that two viewers watch the same video. The first one experiences frequent bitrate changes between 1 Mb/s and 3 Mb/s during playback and the average played bitrate is 2 Mb/s. On the other hand, the second viewer watch the video with 2 Mb/s without any bitrate changes. The average played bitrate is the same. How does this difference affect the viewing experience? Based on the experimental results in Table VII, we confirm that viewers prefer high starting bitrates with no bitrate changes.

C. *Regression analysis*

Throughout the previous experimental results, we find that viewers experience interruptions differently depending on total playback time, rebuffering duration, number of rebufferings and bitrate changes during playback. Based on our YouSlow dataset, we conduct multiple linear regression analysis to investigate the relationship between the abandonment rate and the two viewing interruptions, rebufferings and bitrate changes. To concentrate on the impact of rebufferings and bitrate changes, we omit the video sessions that experienced pre-roll ads and long initial buffering ($>$ 1 second). To reduce the oscillation due to the viewers who abandoned videos during the beginning of playback, we analyze the sessions where the viewers watched the videos for at least 60 seconds (Figure 13). In the dataset, we found a small number of exceptional cases ($< 1\%$) where the viewers watched the videos to the end even if they suffered from a very long initial buffering or rebufferings throughout the entire playback. We considered these samples as falsely reported and removed them from the dataset. Considering the outliers, Table VIII shows the statistics of video session for our regression analysis.

TABLE VIII: Video session statistics for regression analysis

Term	Min.	Max.
Playback length	60 s	1,000 s
Num. of rebufferings	0	6
Total rebuffering duration	0	25 s
Num. of bitrate changes	0	3

1) *Using the number of rebufferings and bitrate changes:*

Can we find a strong linear relationship between the abandonment rate and the number of rebufferings and bitrate changes? We conduct the regression analysis between the abandonment rates and the two factors. Our experimental results are shown in Table IX. We note that the *Pearson* correlation coefficient of two factors is 0.141 (P-value = 0.531). This indicates that the two factors are not highly correlated. According to the ANOVA (analysis of variance) results in Table IXb, two factors predict the abandonment rate significantly, $F(2, 19) = 41.52$, $p < 0.0001$. The fitted regression model found from the analysis is $(\text{Abandonment rate}) = 0.1821 + 0.0246 * \text{Num. of RBs} + 0.0374 * \text{Num. of BR changes}$. The p-values in Table IXc tell us that the number of rebufferings has more impact on the abandonment rate than the number of bitrate changes. We can see the high value of R-squared, 0.814 of the explained variability in abandonment rate. In general, the higher the R-squared, the better the model fits the data.

TABLE IX: Multiple linear regression analysis on abandonment rate using the number of rebufferings (RBs) and bitrate (BR) changes

Predictor variable	S	R-sq	R-sq (adj)
(1) only	0.047	59.6%	57.6%
(2) only	0.061	32.6%	29.2%
(1) and (2)	0.033	81.4%	79.4%

(1) Num. of RBs (2) Num. of BR changes

(a) Model summary

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	2	0.091	0.045	41.52	< 0.0001
Error	19	0.02	0.001		
Total	21	0.112			

(b) Analysis of variance

Term	Coef	SE Coef	T-value	P-value	VIF
Constant	0.1821	0.018	9.92	< 0.0001	
Num. of RBs	0.0246	0.003	7.06	< 0.0001	1.02
Num. of BR changes	0.0374	0.008	4.71	0.0002	1.02

(c) Coefficients

2) Using the number of rebufferings and rebuffering/bitratehange ratios: We categorize the dataset into several groups depending on three predictors, the number of rebufferings and the rebuffering and bitrate change ratios using Equation 1 and 2. For the rebuffering ratio, we first counted the number of sessions with the same rebuffering ratio and tried to calculate the abandonment rate. However, the total playback time varies between 60 seconds and 1,000 seconds. So, it is difficult to gather a sufficiently large number of samples for each rebuffering ratio for the analysis of the abandonment rate. To address this, we split the dataset into multiple subsets depending on the normalized rebuffering and bitrate change ratios. We divided sessions into 0.1 intervals of rebuffering ratio. For instance, we consider the sessions with the same range of rebuffering ratio between 0.1 and 0.2 as the session that has 0.15 of rebuffering ratio. For the bitrate change ratio, we divided the sessions into 0.2 intervals. Table X shows our regression analysis. Figure 24 presents the fitted lines between the abandonment rate and each predictor. Among the three predictors, the rebuffering ratio has the most impact on the abandonment rate while the bitrate change ratio has the least impact (Table Xa). Using the rebuffering ratio and the number of the rebufferings, we can achieve the highest R-squared value ($R^2 = 0.94$).

D. Summary of key observations

These are the key findings from our QoE experimental results:

Finding 1: Our measurements show that about 10% of viewers abandoned the YouTube videos when the pre-roll ads lasted for 15 seconds (Figure 14).

Finding 2: We observe that viewers are more likely to abandon videos with multiple rebufferings compared to a single rebuffering event even if the rebuffering ratio is the same (Figure 20). We confirm that the initial buffering has more impact on the video abandonment than the rebufferings in the middle of a playback (Figure 21). In addition, we observe that viewers prefer constant bitrate to increasing bitrate during playback, but the abandonment rate is not significantly different (Figure 22 and 23).

TABLE X: Multiple linear regression analysis on abandonment rate using the number of rebufferings (RBs) and rebuffering/bitrate (BR) change ratios

Predictor variable	S	R-sq	R-sq (adj)
(1) only	0.06	91.3%	90.9%
(3) only	0.122	64.3%	62.7%
(1) and (3)	0.049	94.6%	94.0%
(1) and (2)	0.0615	91.4%	90.5%
(1), (2) and (3)	0.05	94.6%	93.7%

(1) RB ratio (2) BR change ratio (3) Num. of RBs

(a) Model summary

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Regression	3	0.872	0.29	115.92	< 0.0001
Error	20	0.05	0.002		
Total	23	0.922			

(b) Analysis of variance

Term	Coef	SE Coef	T-value	P-value	VIF
Constant	0.142	0.033	4.28	0.0004	
RB ratio	2.156	0.212	10.17	< 0.0001	2.13
BR change ratio	0.001	0.027	0.07	0.9483	1.32
Num. of RBs	0.031	0.009	3.43	0.0027	2.04

(c) Coefficients

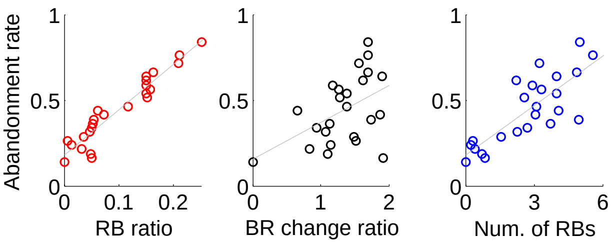


Fig. 24: Fitted lines for multiple linear regression analysis

Finding 3: We find that monitoring the number of rebufferings and the rebuffering ratio is a good metric to quantify video abandonment rates for short videos such as YouTube. Compared to the rebuffering impact, the bitrate change does not affect the video abandonment significantly. To estimate the abandonment rate in YouTube, we can create a strong linear regression model ($R^2 = 0.81$) using the number of rebufferings and bitrate changes only (Table IX). We can increase the R-squared value up to 0.94 with the combination of the rebuffering ratio and the number of rebufferings (Table X).

VI. Discussion

Throughout the previous sections, we describe what YouSlow can measure and how we can analyze video QoE using YouSlow. We believe that our experimental results give us an insight to improving ABR heuristics embedded in ABR players and user engagement for video ads. As we noted, YouSlow platform can be easily implemented for other streaming services such as Netflix[5] and Hulu if they provide any player APIs. In this case, we can monitor and compare behaviors of video watching viewers, between short video clips such as music videos and sports highlights in YouTube and long videos such as movies and TV shows in Netflix [37]. In addition, YouSlow can be used to analyze video ads in social networks. For example, Facebook recently began allowing

[5] As of Nov. 14th, 2014, public API developers are no longer able to access Netflix content. Netflix is taking its API private [51].

embedded videos to play automatically when users scroll to that page. Using YouSlow, we expect to investigate the impact of video ads on user behavior in Facebook and compare the results with YouTube. Furthermore, we can compare video abandonment when viewers watch videos on desktops or mobile devices. Due to lack of reports from mobile devices, we leave this for future work.

VII. Related work

Video QoE analysis: Dobrian et al. [37] at Conviva monitored user-engagement based on various playback events measured from video players. The methodology used for data collection is similar to our approach. They focused on the analysis of initial buffering and rebuffering ratio, not bitrate switching. They found that the rebuffering ratio has the largest impact on video abandonment and the impact is quantitatively different depending on content types. They argued that initial buffering has significantly lower impact on video abandonment, diverging from our findings as shown in Figure 21. Unlike their approach, our platform allows viewers and video service providers to monitor various playback statistics in real time via our QoE monitoring system. In addition, we suggest simpler metrics (e.g., monitoring rebuffering ratio, number of rebufferings, bitrate change ratio over playback time) that can be implemented at video players to estimate abandonment rates. Shafiq et al. [39] monitored video abandonment by inspecting video packets from the ISPs' viewpoint, but the method is more complicated compared to our web browser plug-in that can detect such abandonments directly from within video players. Hossefeld et al. [43] investigated the impact of rebuffering patterns (i.e., how many and often rebufferings appear during playback) on video QoE. They found that it is difficult to estimate video QoE by considering the total rebuffering duration only. Krishnan et al. [40] investigated the effectiveness of video ads by monitoring their completion and abandonment rates. They found that an ad for long-term videos such as TV shows and movies is about 4% more likely to complete than the same ad for the short-term videos such as YouTube's video clips. They also observed that the viewers abandoned more quickly in the beginning of the ad and abandoned at slower rates as the ad progressed.

Collecting measurement data from a web browser plug-in: For analyzing network performance issues such as page loading times, Dhawan et al. [52] introduce *Fathom*, a browser-based network measurement platform. As a proof of concept, they have built a Firefox plug-in that allows web sites or other parties to program network measurements using JavaScript. Barbara et al. [53] have built a YouTube monitoring tool (YoMo) that analyzes the amount of playtime buffered by the YouTube player. The Firefox plug-in focuses on the Flash-based streaming in YouTube and monitors TCP flows at the client in order to estimate the time when the YouTube player is stalling. They focused on the analysis of buffering status of YouTube player, but did not investigate QoE metrics in video streaming.

VIII. Conclusions

We introduced YouSlow as a new video QoE analysis tool for video QoE. Our experimental results show that monitoring the rebuffering ratio and counting the number of rebufferings during playback are proper QoE metrics to analyze abandonment rates for short videos such as YouTube. As key observations, we find that about 10% of viewers closed the videos during the pre-roll ads when the ads lasted for 15 seconds. Further, our analysis shows that viewers prefer constant bitrate to increasing bitrate during playback. Our regression analysis shows that the rebuffering ratio has the most significant impact on the abandonment rate compared to the bitrate change ratio and the number of rebufferings. We believe that our proposed QoE metrics and experimental results give us an insight to improving ABR heuristics in ABR players and enhancing viewing experiences.

Acknowledgment

This work is supported by the National Science Foundation under grant number 12-18977.

References

[1] R. Pantos and J. William May, "HTTP Live Streaming," IETF Draft, Apr. 2015.

[2] Microsoft IIS Smooth Streaming. [Online]. Available: http://www.iis.net/downloads/microsoft/smooth-streaming

[3] Adobe HTTP Dynamic Streaming. [Online]. Available: http://www.adobe.com/products/hds-dynamic-streaming.html

[4] S. Lederer, C. Müller, and C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset," in *Proceedings of ACM MMSys*, Chapel Hill, North Carolina, USA, Feb. 2012.

[5] M. C. Thornburgh, " Adobe's Secure Real-Time Media Flow Protocol," IETF Draft, Jul. 2013.

[6] H. Schulzrinne, A. Rao, R. Lanphier, M. Westerlund, and M. Stiemerling, "Real Time Streaming Protocol 2.0 (RTSP)," IETF Draft, Feb. 2014.

[7] Adobe Media Server family. [Online]. Available: http://www.adobe.com/products/adobe-media-server-family.html

[8] Wowza media systems. [Online]. Available: https://www.wowza.com/

[9] Dynamic RTMP Streaming by JW player. [Online]. Available: https://support.jwplayer.com/customer/portal/articles/1430398-dynamic-rtmp-streaming

[10] An Overview of Digital Rights Management. [Online]. Available: http://www.encoding.com/digital-rights-management-drm/

[11] W3C - Media Source Extensions. [Online]. Available: http://w3c.github.io/media-source/

[12] Technical Note TN2224: Best Practices for Creating and Deploying HTTP Live Streaming Media for the iPhone and iPad. [Online]. Available: https://developer.apple.com/library/ios/technotes/tn2224/_index.html

[13] IIS Smooth Streaming Technical Overview. [Online]. Available: http://www.bogotobogo.com/VideoStreaming/Files/iis8/IIS_Smooth_Streaming_Technical_Overview.pdf

[14] Apple HTTP Live Streaming. [Online]. Available: https://developer.apple.com/streaming/

[15] Source code of Silverlight extensions. [Online]. Available: http://slextensions.codeplex.com/SourceControl/latest#trunk/SLExtensions/AdaptiveStreaming/Heuristics/NetworkHeuristicsParams.cs

[16] Vimeo Player API. [Online]. Available: https://developer.vimeo.com/player

[17] Chrome extensions. [Online]. Available: https://developer.chrome.com/extensions

[18] YouTube player APIs. [Online]. Available: https://developers.google.com/youtube/iframe_api_reference

[19] K. M. Chan and J. Y. B. Lee, "Improving adaptive http streaming performance with predictive transmission and cross-layer client buffer estimation," *Multimedia Tools and Applications*, vol. 75, no. 10, pp. 5917–5937, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11042-015-2556-y

[20] YouTube video ads. [Online]. Available: https://support.google.com/displayspecs/answer/6244541?hl=en&rd=2&ref_topic=6244532

[21] Non-skippable in-stream ads in YouTube. [Online]. Available: https://support.google.com/youtube/answer/188038?hl=en

[22] Chrome and Safari AdBlock. [Online]. Available: https://getadblock.com/

[23] Chrome extensions - webRequest API. [Online]. Available: https://developer.chrome.com/extensions/webRequest

[24] YouTube live encoder settings, bitrates and resolutions. [Online]. Available: https://support.google.com/youtube/answer/2853702?hl=en

[25] R. K. P. Mok, E. W. W. Chan, and K.-C. Chang, "Measuring the Quality of Experience of HTTP Video Streaming," in *Proceedings of IFIP/IEEE IM*, Dublin, Ireland, May 2011.

[26] K.-T. Chen, C.-J. Chang, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "Quadrant of Euphoria: A Crowdsourcing Platform for QoE Assessment," *Journal of IEEE Network*, vol. 24, no. 2, pp. 28–35, March 2010.

[27] P. Casas, R. Schatz, and T. Hossfeld, "Monitoring YouTube QoE: Is Your Mobile Network Delivering the Right Experience to your Customers?" in *Proceedings of IEEE WCNC*, Sanghai, China, April 2013.

[28] Google Video Quality Report. [Online]. Available: http://www.google.com/get/videoqualityreport/

[29] Netflix ISP speed index. [Online]. Available: https://ispspeedindex.netflix.com/

[30] ITU-T P.10/G.100 Amendment 1: New Appendix I Definition of Quality of Experience (QoE). [Online]. Available: https://www.itu.int/rec/T-REC-P.10-200701-S!Amd1

[31] ITU-T Recommendation E.800 Definitions of terms related to quality of service. [Online]. Available: http://www.itu.int/rec/T-REC-E.800-200809-I

[32] ITU-T Recommendation BT.500-13 Methodology for the subjective assessment of the quality of television pictures. [Online]. Available: https://www.itu.int/rec/R-REC-BT.500-13-201201-I/en

[33] ITU-T P.910 Subjective video quality assessment methods for multimedia applications. [Online]. Available: https://www.itu.int/rec/T-REC-P.910-200804-I/en

[34] ITU-T G.107 The E-model: a computational model for use in transmission planning. [Online]. Available: https://www.itu.int/rec/T-REC-G.107

[35] ITU-T P.862 Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. [Online]. Available: http://www.itu.int/rec/T-REC-P.862

[36] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, "Quantifying Skype User Satisfaction," in *Proceedings of ACM SIGCOMM*, Pisa, Italy, Sep. 2006.

[37] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the Impact of Video Quality on User Engagement," in *Proceedings of ACM SIGCOMM*, Toronto, Ontario, Canada, Aug 2011.

[38] Y. Chen, K. Wu, and Q. Zhang, "From QoS to QoE: A Tutorial on Video Quality Assessment," *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 1126–1165, Secondquarter 2015.

[39] 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*, Austin, Texas, USA, Jun 2014.

[40] S. S. Krishnan and R. K. Sitaraman, "Understanding the Effectiveness of Video Ads: A Measurement Study," in *Proceedings of ACM IMC*, Barcelona, Spain, Oct. 2013.

[41] 3 Things You Need to Know About Making a Marketing Video Convert. [Online]. Available: http://www.sailthru.com/marketing-blog/3-things-need-know-making-marketing-video-convert/

[42] A. Floris, L. Atzori, G. Ginesu, and D. D. Giusto, "QoE Assessment of Multimedia Video Consumption on Tablet Devices," in *Proceedings of IEEE Globecom QoEMC Workshop*, Anaheim, California, Dec 2012.

[43] T. Hossfeld, D. Strohmeier, A. Raake, and R. Schatz, "Pippi Longstocking Calculus for Temporal Stimuli Pattern on YouTube QoE: $1+1=3$ and $1\cdot4\neq4\cdot1$," in *Proceedings of ACM MMSys MoVid Workshop*, Oslo, Norway, Feb 2013.

[44] T. Hossfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial Delay vs. Interruptions: Between the Devil and the Deep Blue Sea," in *Proceedings of IEEE QoMEX*, Melbourne, Australia, July 2012.

[45] T. D. Pessemier, K. D. Moor, W. Joseph, L. D. Marez, and L. Martens, "Quantifying the Influence of Rebuffering Interruptions on the User's Quality of Experience During Mobile Video Watching," *Broadcasting, IEEE Transactions on*, vol. 59, no. 1, pp. 47–61, March 2013.

[46] L. Yitong, S. Yun, M. Yinian, L. Jing, L. Qi, and Y. Dacheng, "A Study on Quality of Experience for Adaptive Streaming Service," in *Proceedings of IEEE ICC*, Budapest, Hungary, June 2013.

[47] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH: A QoE-aware DASH System," in *Proceedings of ACM MMSys*, Chapel Hill, North Carolina, Feb. 2012.

[48] A. Rehman and Z. Wang, "Perceptual Experience of Time-varying Video Quality," in *Proceedings of IEEE QoMEX*, Klagenfurt, Austria, July 2013.

[49] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Flicker Effects in Adaptive Video Streaming to Handheld Devices," in *Proceedings of ACM Multimedia*, Scottsdale, Arizona, USA, Nov. 2011.

[50] D. C. Robinson, Y. Jutras, and V. Craciun, "Subjective Video Quality Assessment of HTTP Adaptive Streaming Technologies," *Bell Labs Technical Journal*, vol. 16, no. 4, pp. 5–23, 2012. [Online]. Available: http://dx.doi.org/10.1002/bltj.20531

[51] Netflix is shutting down its public APIs. [Online]. Available: https://gigaom.com/2014/11/14/netflix-is-shutting-down-its-public-api-today/

[52] M. Dhawan, J. Samuel, R. Teixeira, C. Kreibich, M. Allman, N. Weaver, and V. Paxson, "Fathom: A Browser-based Network Measurement Platform," in *Proceedings of ACM IMC*, Boston, Massachusetts, USA, Nov. 2012.

[53] B. Staehle, M. Hirth, R. Pries, D. Staehle, B. Staehle, M. Hirth, R. Pries, and D. Staehle, "YoMo: A YouTube Application Comfort Monitoring Tool," 2010. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.163.3983

Hyunwoo Nam received Ph.D. in Columbia University's electrical engineering department in October 2016. Prior to beginning his studies at Columbia University, he served in the Republic of Korea Army as a communications officer developing a military ad hoc networking system. In 2010, he was a research intern at Metropolitan Transportation Authority (MTA), New York, for a project providing wireless services to passengers on the MTA subway using mesh networks. His collaboration with Bell Labs and Verizon focused on capacity planning for wireless networks, analysis of adaptive bitrate streaming technologies and future networks such as SDN and NFV for intelligent content delivery over wireless.

Henning Schulzrinne is Julian Clarence Levi Professor of Computer Science at Columbia University, received his Ph.D. from the University of Massachusetts in Amherst, Massachusetts. He was an MTS at AT&T Bell Laboratories and an associate department head at GMD-Fokus (Berlin), before joining the Computer Science and Electrical Engineering departments at Columbia University. He served as chair of the Department of Computer Science from 2004 to 2009, as Engineering Fellow at the US Federal Communications Commission (FCC) in 2010 and 2011, and as Chief Technology Officer and Technical Advisor at the FCC from 2012.

He has published more than 250 journal and conference papers, and more than 70 Internet RFCs. Protocols co-developed by him, such as RTP, RTSP and SIP, are now Internet standards, used by almost all Internet telephony and multimedia applications. His research interests include Internet multimedia systems, ubiquitous computing, and mobile systems.

He is a Fellow of the ACM and IEEE, has received the New York City Mayor's Award for Excellence in Science and Technology, the VON Pioneer Award, TCCC service award, IEEE Region 1 William Terry Award for Lifetime Distinguished Service to IEEE, the UMass Computer Science Outstanding Alumni recognition and is a member of the Internet Hall of Fame.