Measuring Decentralized Video Streaming: A Case Study of DTube

Trinh Viet Doan, Tat Dat Pham, Markus Oberprieler, Vaibhav Bajpai
Technical University of Munich, Germany
[trinhviet.doan | dat.pham | markus.oberprieler | vaibhav.bajpai]@tum.de

***Abstract*—The majority of Internet traffic originates from a handful of large-scale video streaming services such as YouTube or Netflix, which continue to grow rapidly. These trends result in traffic and content centralization, which have raised increased awareness of the importance of more decentralized solutions to promote a more open Internet. While P2P-based, decentralized video streaming has been proposed and studied in the past, more novel solutions have not been investigated yet. One recent example is *DTube*, a video streaming platform which leverages decentralized technologies for content delivery and curation. In this study, our goal is to assess the viability of novel decentralized video streaming services, using DTube as an example. We develop and present an Android application to measure connectivity and performance of both DTube and YouTube over WiFi and cellular networks. We collect measurements over a period of 10 months (2019) and from four different cellular ISPs across three different countries. Analyzing the results, we find that streaming videos from DTube is by and large comparable to streaming from YouTube, although we notice a lack of geographically distributed DTube servers and caches. Nevertheless, the results indicate that decentralized video streaming may present a feasible alternative to centralized services in the future.**

I. Introduction

With the number of mobile devices continuously increasing and mobile traffic being dominated by video [1], video streaming accounts for most of the Internet traffic [2] as of 2020. Popular streaming services such as YouTube or Netflix [3] operate a large number of servers in edge networks around the globe in order to bring content closer to the user and to optimize performance [4], [5]. To this end, they also closely cooperate with both fixed-line and cellular ISPs by deploying content caches within ISP networks [6]. Although their infrastructure is globally distributed, these streaming services are managed in a centralized manner, i.e., they have primary control in decision making regarding service operation. Due to their size, they further make it difficult for new services to emerge, while simultaneously increasing their own content variety and availability.

This increasing centralization has raised concerns within different groups of stakeholders on the Internet. Recent efforts [7], [8] advocate the openness and freedom of the Internet and the Web, supporting the development of decentralized applications and services. Decentralized and P2P-assisted video streaming services have been proposed in the past, however, have had difficulties gaining critical mass due to a variety of challenges such as limited access capacity, peer dynamics and heterogeneity, and lack of incentives [9], [10]. An example for a recent approach that tries to tackle these challenges is *DTube* [11]; it leverages decentralized solutions such as the *InterPlanetary File System (IPFS)* [12] to deliver videos from a decentralized P2P network while also using the *Steem* blockchain [13], [14] for content curation and incentivization (§ II). DTube further provides user interfaces and features (e.g., for user interaction and content monetization) similar to YouTube and is still under active development, making it a promising and ready-for-use decentralized alternative.

However, due to the novelty of the involved technologies, such decentralized video streaming ecosystems have not been extensively studied yet—a gap which we would like to fill. To this end, we pose the following research questions: *How does decentralized video streaming compare with centralized services in terms of performance? How distributed are such decentralized services? In which areas can decentralized video streaming be improved?* In order to investigate these questions, we develop an Android application that measures video streaming from YouTube and DTube and analyze the collected measurements. To the best of our knowledge, we are the first to study decentralized video streaming from mobile devices in comparison with an established, centralized video streaming solution. Our main contributions are:

- We implement an Android app that can be used to perform measurements of video streaming services and open-source the code. The app measures metadata as well as connectivity-related metrics of streamed videos from YouTube and DTube, applying the same methodology (§ III) in terms of playout logic to obtain comparable measurements. Thus, the app can be extended to measure other video streaming platforms for future studies.
- We perform measurements for more than 8,500 videos from YouTube and DTube over WiFi and cellular networks across a period of 10 months (February until November 2019). The collected data includes measurements from DE, CZ, and the US, as well as measurements from four different cellular ISPs over LTE (§ III-D).
- We analyze the measurements (§ IV) and notice YouTube to generally provide better performance, although measurements for DTube are not substantially worse. Over the study period, we observe similar and stable behavior across all months for both platforms, showing no significant longitudinal change.

ISBN 978-3-903176-28-7

TCP connect times are in the same order of magnitude (< 100 ms) for both services, however, are only about half as long for YouTube. Moreover, we find startup delays of videos to be higher by a factor of four for DTube videos (3.2 to 0.82 seconds over WiFi, 5.8 to 1.35 seconds over cellular). We determine paths toward YouTube to be shorter than to DTube by 7–8 IP hops, as YouTube operates servers and caches in edge networks all around the globe, whereas we see all DTube measurements accessing servers located in FR. As such, playout of DTube videos exhibits particularly bad performance when inspecting the measurements performed from the US. This indicates a lack of globally distributed servers for DTube as of 2019.
Overall, this observation suggests that, although DTube leverages decentralized technologies, its infrastructure is still geographically centralized, which represents a limiting factor for streaming performance. Nevertheless, we find DTube to be able to perform at a level below but comparable to YouTube, albeit only in cases where infrastructure is deployed and accessible close to the user. These findings underline the potential of decentralized video streaming as an alternative to established, state-of-the-art centralized video streaming services in the future.

To encourage future work in the domain of measuring decentralized video streaming, we release the source code of the app (which we still continue to develop and improve) to the public[1]. The app can be extended to cover additional streaming services, so that measurements (from the user perspective) can be collected for various purposes. Moreover, we make the collected measurements and analysis scripts used in this study available[2] to facilitate reproducibility [15].

II. Background and Related Work

A. Background

1) DTube / Steem: DTube [11] is an open-source, decentralized video streaming project that is built on top of a P2P file system (*IPFS*, see § II-A2) [12] and leverages the *Steem* [13] blockchain. Similar to *Steemit* [16], [17], [18], an ecosystem for social media and news aggregation on top of the Steem blockchain, DTube allows user-created content to be curated by other users. Further, storing the metadata and curation information in the Steem blockchain makes content resistant to censorship. Through the integration of the Steem blockchain, it also taps into the active Steem user base with more than 1.3M registered accounts [14]. *Steem Dollars* are a cryptocurrency, given out and exchanged among users to add monetary value to content (videos, comments, ...), incentivizing users to benignly participate and share content on the platform. DTube videos are delivered through a distributed file system, the InterPlanetary File System (IPFS) [12], [19], a P2P protocol stack for file exchange. In the case of Youtube, both content as well as personal and meta data are stored centrally on Google servers; DTube aims to prevent having a single point of control by using the aforementioned decentralized solutions instead. Moreover, DTube mimics many features of YouTube in order to provide a familiar user experience. Other novel decentralized video streaming services such as *PeerTube* or *Viewly* lack user-interaction features or incentives, which is why we consider DTube to be the closest replica to YouTube (as of 2019), making it the focus of our study. Note that DTube has undergone several changes since early 2019, in particular support for additional video sources (from other P2P networks and centralized third parties); however, our study is limited to IPFS, which was used by DTube initially.

2) IPFS: The InterPlanetary File System [12] combines a set of approaches from communication and networking research as well as file management into a protocol stack to build a distributed file system on top of a peer-to-peer (P2P) network. Its goal is to provide a "distributed and permanent Web" and an alternative to HTTP, which has become the de facto standard for new applications and services despite its multiple limitations [20]. IPFS tries to mitigate some of the limitations by applying concepts of Information-Centric Networking (ICN), using uniquely identifying fingerprints to address and retrieve files over the P2P network from any peer (i.e., information-centric), rather than downloading from a fixed location which is given by a server address (host-centric). Each file is further chunked into uniquely fingerprintable pieces, with the chunks' hashes being organized in a Merkle tree. Peers can then find and exchange these chunks via the fingerprint, akin to *BitTorrent*, and verify the integrity of the whole file by reconstructing the Merkle tree in the end.

To simplify the file retrieval process, peers can act as a Web browser-accessible gateway to the IPFS network: This allows external users (which do not run an IPFS node themselves) to simply request and retrieve content from IPFS via HTTP, with the gateway node carrying out the retrieval process outlined above on their behalf. DTube provides its own IPFS gateway at `video.dtube.top`, which users and our measurement app stream the videos from, and which we also refer to as DTube server(s) for the remainder of this study. However, when users upload a video via DTube's Web interface, the video is only stored in its private/sandboxed IPFS network, which is only accessible through the mentioned gateway and is isolated from the public (default) IPFS network. However, DTube also allows users to share and embed videos from the public IPFS network through their IPFS hashes.

B. Related Work

There have been plentiful measurement studies on Video on Demand (VoD) services, with focus on Quality of Service (QoS), Quality of Experience (QoE), and other network-level metrics [21]. Previous work has extensively studied YouTube as an example for a platform that allows streaming of VoDs. Studies have specifically investigated streaming YouTube videos from mobile devices in the recent past: *YoMoApp* [22], a monitoring mobile application for Android, measures performance indicators for YouTube's HTTP adap-

[1] https://github.com/tv-doan/ifip-net-2020-app
[2] https://github.com/tv-doan/ifip-net-2020-analysis

tive streaming [23]. Longitudinal data collected by the app over five years reveals that YouTube and cellular networks have improved performance and QoE systematically for mobile devices [24]. Other studies investigate YouTube streaming with respect to IPv6 delivery [25], load balancing [26], and content cache deployments [6]. All of these studies have performed measurements specific to YouTube, a centrally operated video streaming service, which indicates a gap with respect to understanding decentralized solutions.

Studies investigated P2P-based video streaming in the past [27], [28], [29], [9], [30], [10], primarily before 2010, and found that P2P-based solutions had promising opportunities, e.g., to reduce bandwidth costs. However, these solutions also faced challenges such as bottlenecks due to flash crowds, inefficient handling of the number of stream channels, and other peer population dynamics. Nevertheless, it is unclear whether current infrastructure and technologies are able to overcome these challenges nowadays, as more recent decentralized video streaming services have not been extensively studied yet. More recent studies [31], [32] have seen that peer-assisted video streaming concentrates interests and access patterns, indicating centralization despite a decentralized architecture. However, the studies do not compare the performance of such decentralized services with centralized ones. Similar findings concerning centralization around a small number of instances, content categories, and ASes have also been revealed by Raman *et al.* [33] for *Mastodon*, a decentralized social networking service. Due to running an isolated IPFS network only accessible through its specific gateway, DTube may also be affected by such patterns.

Overall, new decentralized solutions and ecosystems are increasingly proposed, although the understanding of their feasibility, performance, scalability, and other properties is lacking. To help filling these gaps and to motivate further research in this area, we conduct an empirical study in which we measure and compare centralized with decentralized video streaming by example of YouTube and DTube. We acknowledge that both platforms have inherently different resources and infrastructure, which biases the comparison. However, from a user perspective, these differences are not visible and only reflected in the services' performances.

III. Methodology

A. Overview

We develop a mobile app to measure the connectivity and performance of YouTube and DTube, and open-source the code. Due to mobile devices gaining popularity, we develop the app for Android, the most popular mobile operating system. Currently, the app only supports measuring VoDs from YouTube and DTube (via IPFS). Fig. 1 presents a high-level overview of the measurement process. The area shaded in gray, which represents the extraction of the video source URLs embedded in the video webpages, can be adapted to extend the app to capture other VoDs streaming services. The application logic can also be integrated in other modules that already include video playout features.

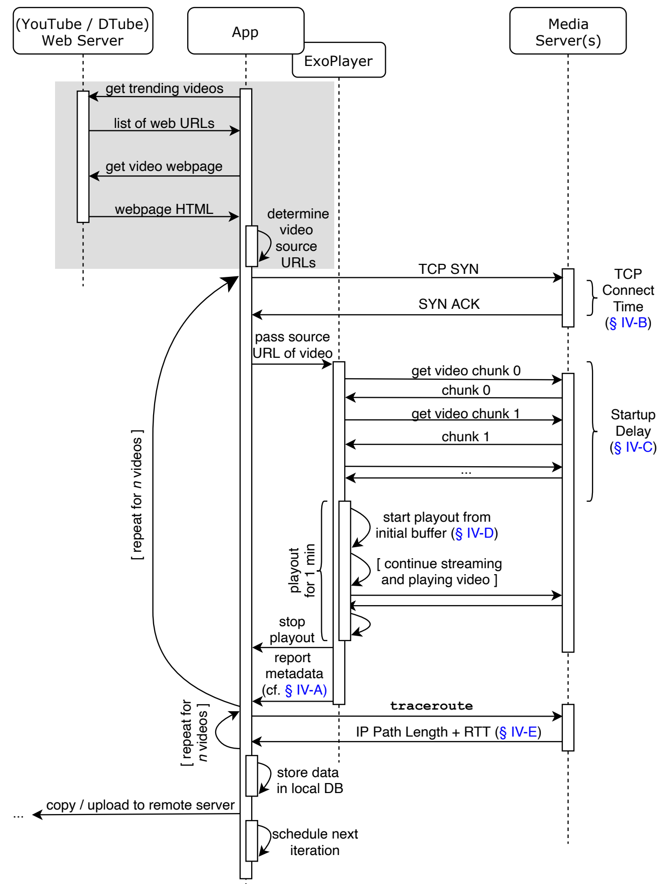

Fig. 1. Sequence diagram of high-level measurement process (timings not proportional). The area shaded in gray follows a different approach on the lower level depending on the platform, however, is the same on a more abstract level (see § III-B). Note that the app collects additional metrics and (meta)data, such as processing time or network type, which are not shown.

To reduce the interference of the measurement app with regular usage of the mobile device, we implement the measurement as a foreground service. The service simulates real use scenarios of video streaming on a mobile device, i.e., it actually streams and plays the video (either in foreground or background). For the media download and playout, we leverage Google's `ExoPlayer`, which is a customizable application-level media player and is also used by the official YouTube Android app, for instance.

B. Measurement Process

For each measuring cycle, the *trending* list of videos on each platform is retrieved. From each trending list, n random videos (with n being a number specified by the user in the app) are selected to be measured. YouTube offers an API to retrieve the webpage URLs of these videos, however, an additional open-source library (`android-youtubeExtractor`) is used to determine the source URL of the video, which is passed to the `ExoPlayer` for streaming. DTube, on the other hand, does not offer a similar API; thus, we implement a workaround that loads and parses the respective pages via Android `WebView`, a Chromium-based component that retrieves and displays Web content, in order to retrieve the webpage and source URLs (pointing to a DTube gateway) of the videos. Note that we find many videos served by the DTube gateway to not load properly, see § III-D. Further, these steps depend on the API and frontend of the services, which might change in the future.

As the default resolution of videos on DTube is 480p, we choose and stream videos in 480p on both platforms to collect comparable measurements. We do not consider adaptive bitrate streaming, as this is not supported by DTube. We acknowledge that different resolutions may result in varying observations. However, we believe that a resolution of 480p is a reasonable compromise in terms of video quality and video size for mobile devices: going below 480p could improve measured performances, however, lower resolutions are considered barely watchable as of 2019.

The app uses a common download and playout logic for videos of both any streaming platform and repeats this for all selected videos, and for both platforms: `ExoPlayer` downloads the video files from the video-hosting server in chunks via the source URL, i.e., streams them. The playout duration is limited to one minute. Studies [34] suggest that the duration of video measurements should be at least one minute, preferably three minutes, to obtain more accurate results; however, we opt for the lower bound to reduce resource consumption. After playout, the app performs an ICMP `traceroute` measurement to the IP address where the video is streamed from to determine the IP path length and RTT. In the last step, the app saves the measured metrics to a SQLite database, which is stored on the phone. Additionally, the database is copied to a remote server, which acts as a collection point for measurements from multiple devices. If the scheduling option is enabled, the full measurement process is automatically repeated after i hours (with i specified by the user in the app) have passed.

C. Measurement Metrics

Before, during, and after playout, the app collects a variety of network-related metrics as well as metadata on the videos: for instance, the content duration (§ IV-A) represents the total duration of the video. TCP connect time (§ IV-B) is the time it takes to connect to the media server. Startup delay (§ IV-C) reports the time it takes from the moment the source URL is passed to the `ExoPlayer` until the video starts playing. The initial buffer size (§ IV-D) is the amount of video content buffered (in seconds) before the video starts playing. After the one minute playout of the video, the app concludes by performing an ICMP `traceroute` measurement (§ IV-E) to the media server. Other metrics are also collected throughout the measurement process; however, some of the collected (QoE) metrics are experimental in the version of the app used for the study, and therefore not further discussed. Although the cryptocurrency-based incentivization of DTube is an important feature, we focus on the connectivity-related measurements in the analysis, as we argue that a performant backend and network architecture should be prioritized over other features.

TABLE I
NUMBER OF COLLECTED CONNECTIVITY/PERFORMANCE MEASUREMENTS (TOP) AND `TRACEROUTE` MEASUREMENTS (BOTTOM).

Video Msm.	WiFi	SIMPLE Mobile (US)	T-Mobile (DE)	Vodafone (DE)	o2 (DE)	All
DTube	1820	200	233	87	474	2814
YouTube	4074	199	462	83	919	5737
All	5894	399	695	170	1393	8551

`traceroute`	WiFi	SIMPLE Mobile (US)	T-Mobile (DE)	Vodafone (DE)	o2 (DE)	All
DTube	1556	114	16	87	417	2190
YouTube	3822	124	15	83	839	4883
All	5378	238	31	170	1256	7073

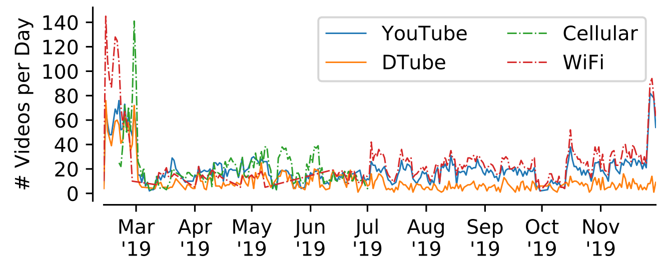

Fig. 2. Number of videos successfully measured per day, split by network type and platform. Cellular measurements are collected until July 2019. The number of DTube videos is lower due to loading errors. Spikes in the beginning and end of the study period are results of configuration changes.

D. Measurement Setup

Over the course of our study period from February 2019 until November 2019 (10 months), we measure over 8,500 videos from both video streaming platforms combined across four mobile phones (two LG Nexus 5X, one Huawei P9, one Xiaomi Mi A1). Table I shows the total count of successful measurements on top, split by platform (YouTube or DTube) and network type/provider (WiFi or cellular ISP). We define a measurement as successful if all metrics have been collected without errors, i.e., have a value ≥ 0. The bottom table shows the number of successful `traceroute` measurements, i.e., traces that have reached the media server and returned within 5 seconds. Note that the distribution is skewed, as roughly twice the amount of YouTube videos have been measured compared with DTube ones. This is due to DTube videos not loading properly from its IPFS network. Moreover, DTube has started to support videos from other sources (e.g., YouTube, Facebook, or other P2P networks) on their platform, meaning that the videos selected for measurements might not be hosted on IPFS, which the app then ignores.

In the beginning of the study (February 2019), we have collected measurements over cellular networks from three different geographical areas over a 1–2 week period: Munich (DE), Prague (CZ), and San Diego (US). However, as we do not observe substantial differences between measurements from DE and CZ, we group these together as EU measurements for the remainder of this study. Further, we find changes over time to be marginal; as a result, we do not discuss results with respect to longitudinal evolution and instead examine the measurements collectively across all months.

We use 4 different SIM card providers for the measurements from cellular networks: T-Mobile (DE), Vodafone (DE), o2 (DE), and SIMPLE Mobile (US). The three former ones operate their own cellular infrastructure; on the other hand, SIMPLE Mobile is a Mobile Virtual Network Operator which uses the infrastructure of T-Mobile in the US. All of the contracts used in our study provide LTE speeds. We use the three DE-based SIM cards for measurements around Munich (T-Mobile around Prague), whereas we use the SIMPLE Mobile card for the measurements in San Diego. Most of the cellular measurements have been collected until July 2019, after which we have switched to WiFi measurements. Fig. 2 visualizes this, showing the number of videos measured per day, split by network type and platform. Measurements over WiFi are collected from the devices in stationary locations (mainly from campus network), while cellular measurements have primarily been collected, with the devices moving within the local areas of each city.

IV. Analysis

A. Content Duration

We begin our analysis by studying and comparing the content duration, i.e., the video length in seconds, of trending videos on both platforms. We observe that trending videos on YouTube tend to be longer than on DTube: The median content duration on YouTube is around 619 seconds, i.e., roughly above 10 minutes. In comparison, the median content duration on DTube is approximately 323 seconds, which is about half of the median of YouTube videos.

We suspect that this is due to the monetization option of YouTube, which allows (and therefore incentivizes) creators to add additional advertisements to a longform video, i.e., a video above 10 minutes [35]. On the other hand, DTube videos are monetized through a user-curated system based on the Steem and DTube blockchain, which decouples video lengths from rewards. Rewards, especially monetary, can be a primary motivation for participation in video streaming ecosystems; while it is important to take user incentives into account when operating a video streaming platform, other aspects such as content variety, user experience, and network and streaming performance also must be considered. We focus on the latter in the following sections.

B. TCP Connect Time

After selecting a video and processing the webpage on which the video is embedded, the app extracts the video source URL and connects to the media server (YouTube media server or DTube IPFS gateway). The app measures the time it takes to establish the TCP connection to the server. The distribution of these TCP connect times by platform and network type is shown in Fig. 3; the distribution split by cellular ISP is shown in Fig. 4 instead.

In general, TCP connect times to YouTube are lower in comparison with DTube over both cellular and WiFi. For instance, the 75th percentile for WiFi measurements to YouTube is at 22 ms (cellular 44 ms); on the other hand, the 75th percentile

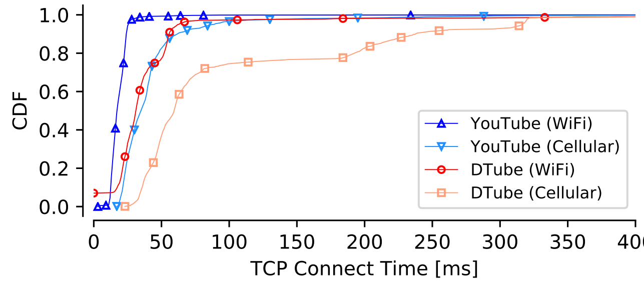

Fig. 3. Distribution of TCP connect times by network type. Establishing connections to YouTube is faster in most cases, although TCP connect times of YouTube over cellular networks and DTube over WiFi are very similar.

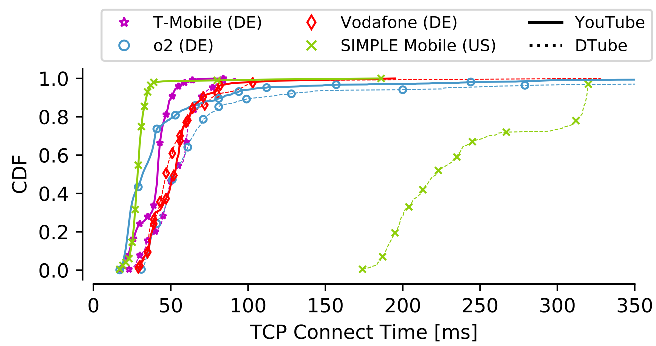

Fig. 4. Distribution of TCP connect times for cellular measurements only (YouTube: solid lines DTube: dashed lines). Measurements from all cellular providers exhibit similar behavior for YouTube, however, measurements to DTube from the US are significantly worse.

is at 45 ms for DTube (cellular 107 ms). This shows that TCP connections to YouTube are established more than twice as fast, although TCP connect times are still within the same orders of magnitude for both platforms. The measurements suggest that TCP connect times to YouTube over cellular are comparable with TCP connect times to DTube over WiFi, while cellular measurements for DTube are trailing behind. However, Fig. 4 shows that the 75th percentile for each ISP to both platforms is within 45–60 ms, with the exception of measurements performed via SIMPLE Mobile to DTube: For SIMPLE Mobile, the 75th percentile is about 300 ms instead, which heavily skews the general distribution shown in Fig. 3. This is explained by YouTube having points-of-presence and caches all around the globe, for instance due to peering of Google with ISPs, which makes YouTube content highly accessible. On the other hand, all DTube servers are located in France (see § IV-E below); consequently, the geographical distance between US and FR alone heavily affects the experienced TCP connect time.

Nevertheless, while TCP connect times to YouTube are generally shorter, the connect times between YouTube and DTube are below roughly 50 ms when servers or caches for both platforms are close by. As such, leveraging content caches, such as additional peers or gateways, that store the requested videos can drastically improve TCP connect times for decentralized video streaming services.

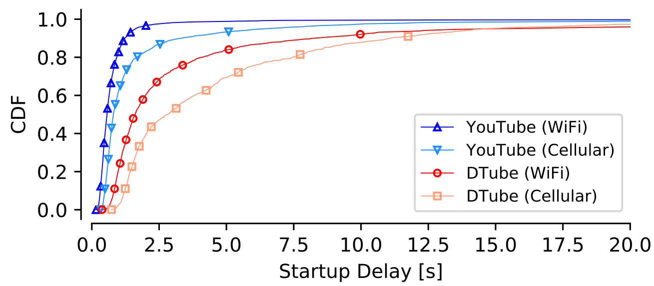

Fig. 5. Distribution of startup delays by network type. The startup delays for YouTube are much lower than for DTube.

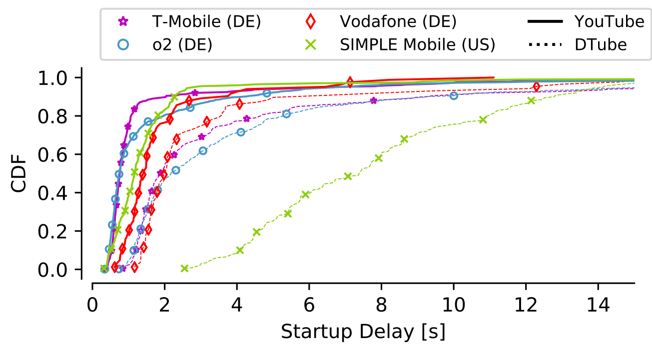

Fig. 6. Distribution of startup delays for cellular measurements only (YouTube: solid lines, DTube: dashed lines). Similar to TCP connect times, measurements to YouTube show conformable behavior, whereas measurements from the US to DTube exhibit much higher startup delays.

Takeaway: We find that TCP connections to YouTube are established in less than half of the time when compared with DTube for the majority of the measurements (75th percentiles for WiFi: 22 to 45 ms; cellular: 44 to 107 ms). In particular, measurements collected from the US via SIMPLE Mobile show very high discrepancies between the both platforms (30 to 300 ms).

C. Startup Delay

To get a more accurate representation of user experience, the app also measures the startup delay, i.e., the time it takes from the video player (`Exoplayer`) processing the video source URL until the actual playout of the first video frames starts.

While startup delays in real scenarios might be different from the ones we measure (due to media player and app optimizations), this does not affect the comparability of the measurements: The video processing and playout logic of the app is platform-agnostic; the measurements that we have collected share the same measurement framework, and are therefore comparable.

Fig. 5 presents the overall distribution of measured startup delays between network types and video platform; Fig. 6 splits the cellular measurements by ISP, in addition. We notice that the startup delay for DTube is much higher when compared with YouTube over both WiFi and cellular. Overall, we see characteristics which are similar to the one we observe for TCP connect times (see § IV-B).

For instance, measurements over WiFi are lower in comparison with cellular ones for the same platform. 75% of the measured YouTube videos require up to 0.82 seconds to start, whereas 75% of the DTube videos require up to 3.2 seconds instead. Over cellular, the difference becomes much larger, as the 75th percentile is 1.35 seconds for YouTube and 5.8 seconds for DTube regarding the startup delay, i.e., a difference of roughly 4.5 seconds.

Splitting the measurements by cellular ISP, we again observe that the cellular measurements toward DTube from the US via SIMPLE Mobile (75th percentile: 9.8 seconds) perform significantly worse than other ISPs. More specifically, the 75th percentiles of all other providers range from 3.1–4.6 seconds for DTube, while for YouTube, all providers (including SIMPLE Mobile) range from 1–1.8 seconds. As such, the startup delays observed for DTube are more than double the ones observe for YouTube, showing much room for improvement. Likely, YouTube performs better due to more content servers and caches deployed globally. On the other hand, DTube lacks a dedicated and distributed infrastructure as of 2019, which negatively impacts the startup delay.

However, conceptually, the distribution of a file on IPFS grows together with its popularity. As such, a greater number of participating IPFS peers might improve this lack of geographical distribution, although a higher number of peers comes with a greater overhead as well, which may cause scalability issues and thus requires follow-up studies on IPFS. Moreover, the isolated IPFS network used by DTube might also exhibit a different behavior regarding file retrieval in comparison with files retrieved from the public IPFS network.

Takeaway: Startup delays for DTube are much higher in comparison with YouTube; considering the 75th percentiles, startup delays for DTube are about four times higher (WiFi: 3.2 to 0.82 seconds, cellular: 5.8 to 1.35 seconds). Similar to TCP connect times, we find the highest differences for measurements performed in the US (9.8 to 1.6 seconds).

D. Initial Buffer Size

We further capture the initial buffer size (in seconds of playable video content) at the moment our app starts playing the video, i.e., when `ExoPlayer` switches its state to `STATE_READY`. We set the buffer required for playback to at least 2 seconds. Thus, this metric heavily depends on the connection to the video source as well as the download speed and throughput. However, as we measure the videos back to back on the same device with no interference, the download speed should be similar, meaning that variation in download speed is only a minor factor of the initial buffer size. Fig. 7 shows the distributions of the initial buffer sizes by platform and network type, with Fig. 8 showing the distributions split by cellular ISP in addition. Both platforms have similar initial buffer sizes when comparing their WiFi and cellular measurements with each other. The 75th percentiles of YouTube are 8.1 seconds for both WiFi and cellular. In comparison, the 75th percentiles are 3.7 seconds (WiFi) and 3.0 seconds (cellular) for DTube,

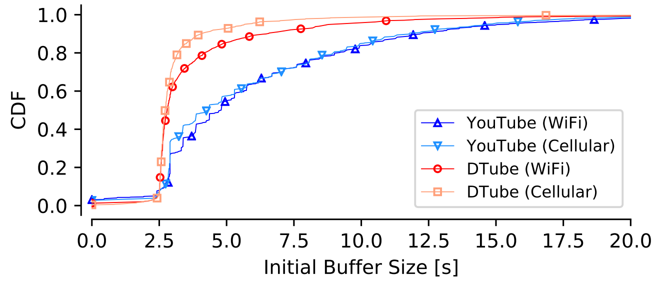

Fig. 7. Distribution of initial buffer sizes by network type. Buffers for YouTube are larger than for DTube (75th percentile: 8.1 vs 3.7 seconds)

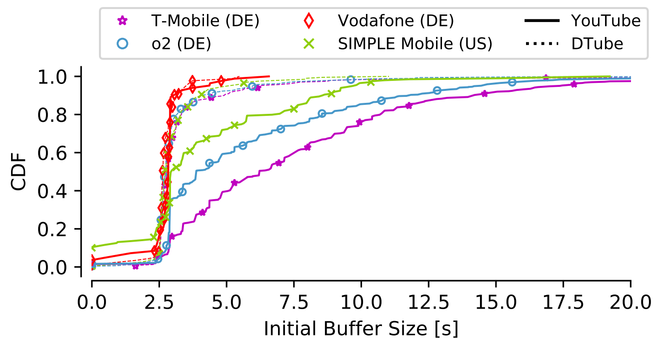

Fig. 8. Distribution of initial buffer sizes for cellular measurements only (YouTube: solid lines, DTube: dashed lines). Buffer sizes for DTube are similar across all providers and vary between different providers for YouTube.

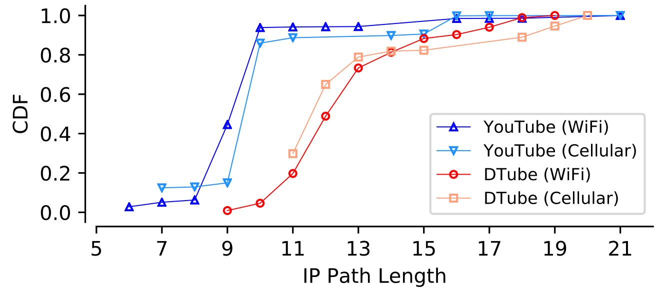

Fig. 9. Distribution of IP path lengths by network type. IP paths to YouTube are shorter by around at least 2 hops in comparison with DTube path lengths.

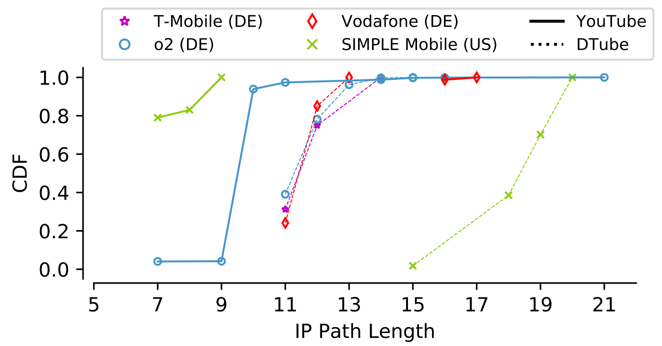

Fig. 10. Distribution of IP path lengths for cellular measurements only (YouTube: solid lines, DTube: dashed lines). The measurements from the US to DTube are by far the largest in terms of IP path lengths, especially compared their YouTube counterpart.

showing that initial buffers are filled with much less video content when streaming from DTube.

When we take the cellular ISPs in consideration, we find substantial differences among the ISPs for YouTube: the 75th percentiles of the buffer sizes range from 2.9 seconds (Vodafone) over 5.6 seconds (SIMPLE Mobile) and 7.6 seconds (o2) to 9.9 seconds (T-Mobile). On the other hand, the 75th percentiles of buffer sizes for DTube cover a much smaller range with 2.9–3.1 seconds for all providers, despite the startup delays also being longer overall which would allow more frames to be downloaded if conditions were equal. Therefore, we suspect that the buffer sizes are bottlenecked toward the server-side for DTube. For YouTube, buffer sizes appear to be limited by the cellular network and download speed instead, as we observe vastly different buffer sizes for different ISPs.

In § IV-B, we have seen that TCP connect times are closer (relatively) to each other than the initial buffer sizes when comparing the platforms. Due to our measurements from Europe having comparable network conditions for both YouTube and DTube, our results highlight the importance of optimizing the server-side object retrieval and throughput for decentralized streaming services.

E. IP Path Length

After playing the video, the app performs an ICMP `traceroute` measurement toward the server where the video has been streamed from. We capture the IP path length to reach the server (see Fig. 9 and 10), together with the corresponding round-trip time (see Fig. 11 and 12). Note that we only consider successful `traceroute` measurements, i.e., measurements that have reached the destination server as well as returned within 5 seconds. As shown in Table I, only around 7.1k `traceroute` measurements out of the 8.5k measurements overall are successful, i.e., a failure rate of 17.3%. We further notice that T-Mobile and SIMPLE Mobile show a much lower success rate (4.5% and 59.6%, respectively) for `traceroute` measurements in comparison to the other two cellular ISPs Vodafone and o2 ($>$ 90%).

With respect to IP path lengths, we find that most YouTube servers are located within 10 IP hops (Wifi: 93.9%, cellular: 86.0% of measurements). In contrast, only 4.6% of the DTube measurements over WiFi reach DTube servers within 10 IP hops; over cellular, the lowest IP path length we observe is 11 (29.8% of traces). Overall, we see DTube servers to be up to 7–8 IP hops farther than YouTube servers.

While YouTube is known to have servers and caches deployed globally, often even peering with ISPs to bring content to the edge, we notice that all `traceroute` measurements toward DTube end in AS16276, managed by OVH (FR), as opposed to YouTube, for which some traces end in the ISP network instead of Google's AS15169. Although OVH also operates servers around the globe, our previous observations and the `traceroute` measurements highly suggest that the DTube traces end in France primarily. In particular, this is

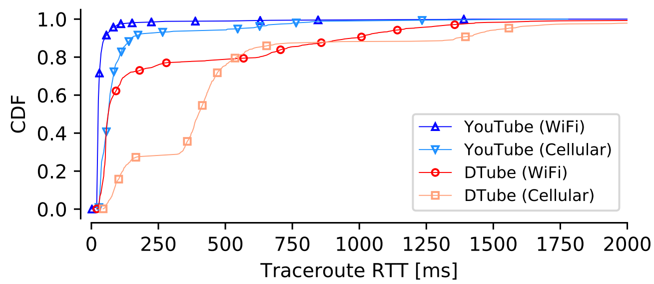

Fig. 11. Distribution of `traceroute` RTTs by network type. While stable RTTs are observed for YouTube, RTTs are more varying to DTube, especially over cellular.

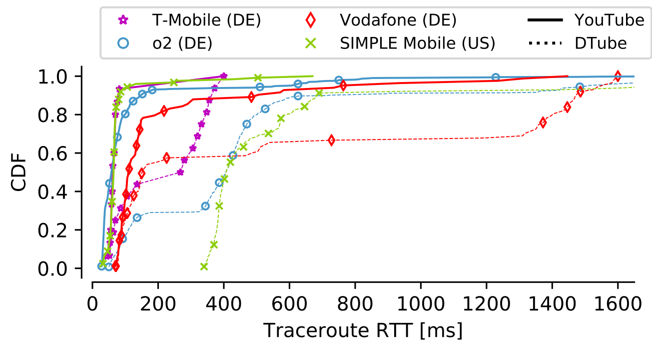

Fig. 12. Distribution of `traceroute` RTTs for cellular measurements only (YouTube: solid lines, DTube: dashed lines). RTTs to YouTube are comparable across providers, whereas for DTube, RTTs exhibit high variance.

reflected in the measurements via SIMPLE Mobile to show the largest differences between YouTube and DTube: Measurements to YouTube indicate the presence of local servers or caches nearby, whereas ones to DTube exhibit much higher timings and indicate oversea connections.

The same pattern is also visible for the IP path lengths when considering individual cellular ISPs (Fig. 10). IP path lengths toward DTube are very similar for all three ISPs from DE (around 11–14 IP hops). However, the US traces range from path lengths of 15–20 hops instead. We further observe a high variance in IP path lengths toward YouTube: Measurements via T-Mobile and Vodafone from DE require around 16 IP hops to reach YouTube servers, whereas ones via o2 (DE) and SIMPLE Mobile (US) only require 9–10 IP hops. This inflation can be seen due to additional internal hops traversed when routing within T-Mobile and Vodafone networks.

Regarding the RTTs (Fig. 11), we see the 66th percentiles of the RTTs to range from 28 ms (YouTube over WiFi) over 76 ms (YouTube over cellular) and 108 ms (DTube over WiFi) to 446 ms (DTube over cellular), indicating that DTube exhibits slow behavior over cellular networks. However, this is only partially reflected in terms of TCP connect times (see § IV-B), for which both YouTube and DTube show less varying and more comparable measurements.

Dissecting cellular measurements into individual ISPs, we find that `traceroute` RTTs for YouTube are rather similar among different providers. For instance, the 80th percentiles are at 80 ms (T-Mobile, SIMPLE Mobile), 100 ms (o2), and 188 ms (Vodafone). However, when examining RTTs for DTube, the values are much more diverging, with the 80th percentiles being 348 ms for T-Mobile, 502 ms for o2, 1404 ms for Vodafone, and 586 ms for SIMPLE Mobile. In particular, we observe the highest DTube RTTs for Vodafone, even though the geographical distance (DE→FR) is much lower when compared with the SIMPLE Mobile traces (US→FR). Although the corresponding IP path lengths support this observation, note that processing of ICMP packets might be delayed in favor of packets with actual payloads. Improving routing as well as having a higher number of content servers around the globe would likely benefit DTube on the path of becoming an alternative, decentralized video streaming platform to YouTube.

Takeaway: With respect to IP path lengths, we see paths to DTube to be up to 7–8 IP hops longer than to YouTube. Around 90% of the YouTube destinations are within 10 IP hops, whereas for DTube, more than 95% of the destinations are beyond 10 IP hops. We discover that all traces toward DTube end in FR; in comparison, traces to YouTube across all measurements primarily end in content caches close by. This also explains the previously discussed outliers for TCP connect time and startup delay measurements from the US.

V. Conclusion and Future Work

We developed an Android application which streams from both a centralized as well as a decentralized video streaming service to measure their connectivity and performance. The app, which we described in terms of design decisions and implementation, uses the same processing and playout logic for both YouTube and DTube, which we picked as representative platforms for the centralized and decentralized streaming services. We presented and compared measurements collected over a period of 10 months from February until November 2019, which include data points from cellular networks in Germany, Czech Republic, and the USA. Regarding content duration, we saw trending videos on YouTube to be longer than on DTube, likely because monetization on YouTube is coupled to video length. Compared with the baseline performance that we defined YouTube as, we noticed DTube to lag behind in many areas, although not by much. For instance, while most TCP connect times were around half for YouTube when compared with DTube (22 to 45 ms over WiFi, 44 to 107 ms over cellular), the majority of the measurements was below 100 ms and within the same order of magnitude. We found more extreme differences for startup delays, for which DTube exhibits values that were around four times higher in comparison with YouTube (3.2 to 0.82 seconds over WiFi, 5.8 to 1.35 seconds over cellular). Further, we observed the initial buffers, i.e., the amount of content buffered before playout of the video, to be smaller for DTube by around 54–63% (3–3.7 seconds for DTube, 8.1 seconds for YouTube over both WiFi and cellular).

Although WiFi and cellular measurements were similar for most metrics, we determined measurements from the US to perform particularly poorly when streaming videos from DTube. Using `traceroute` measurements, we found paths to YouTube to be shorter by around 7–8 IP hops than to DTube (10 vs 17–19 IP hops); in fact, for all of our DTube measurements, we identified the videos to be streamed from servers located in France. This geographical limitation, together with the other results, explained the difference between the EU- and US-based measurements: Even as a video streaming service based on decentralized P2P technologies, DTube lacked the distribution and accessibility of reliable content servers globally, indicating geographical centralization. Despite not having an infrastructure and resources comparable to YouTube, however, DTube exhibits a promising performance for its recent deployment, underlining the potential of decentralized video streaming.

In future work, we will extend the measurement app to include additional metrics related to QoE and streaming sources, e.g., PeerTube or videos from the IPFS network directly. We will also collect more measurements and consider TCP packet traces in order to deepen and normalize our analysis.

References

[1] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, C. Ziemlicki, and Z. Smoreda, "Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage," in *ACM CoNEXT*, 2017. [Online]. Available: https://doi.org/10.1145/3143361.3143369

[2] M. Trevisan, D. Giordano, I. Drago, M. M. Munafò, and M. Mellia, "Five Years at the Edge: Watching Internet From the ISP Network," *IEEE/ACM Transactions on Networking*, 2020. [Online]. Available: https://doi.org/10.1109/TNET.2020.2967588

[3] T. V. Doan, V. Bajpai, and S. Crawford, "A Longitudinal View of Netflix: Content Delivery over IPv6 and Content Cache Deployments," in *INFOCOM*, 2020, [to appear].

[4] Google, "Google Edge Network," 2019, [accessed 2020-Apr-28]. [Online]. Available: https://peering.google.com/#/infrastructure

[5] T. Böttger, F. Cuadrado, G. Tyson, I. Castro, and S. Uhlig, "Open Connect Everywhere: A Glimpse at the Internet Ecosystem through the Lens of the Netflix CDN," *SIGCOMM CCR*, 2018. [Online]. Available: https://doi.org/10.1145/3211852.3211857

[6] T. V. Doan, L. Pajevic, V. Bajpai, and J. Ott, "Tracing the Path to YouTube: A Quantification of Path Lengths and Latencies Toward Content Caches," *IEEE COMMAG*, 2019. [Online]. Available: https://doi.org/10.1109/MCOM.2018.1800132

[7] "Contract for the Web," 2019, [accessed 2020-Apr-28]. [Online]. Available: https://contractfortheweb.org/

[8] Internet Society, "Internet Society 2020 Action Plan," 2019, [accessed 2020-Apr-28]. [Online]. Available: https://www.internetsociety.org/action-plan/2020/

[9] J. Liu, S. G. Rao, B. Li, and H. Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast," *Proceedings of the IEEE*, 2008. [Online]. Available: https://doi.org/10.1109/JPROC.2007.909921

[10] N. Ramzan, H. Park, and E. Izquierdo, "Video Streaming over P2P Networks: Challenges and Opportunities," *Sig. Proc.: Image Comm.*, 2012. [Online]. Available: https://doi.org/10.1016/j.image.2012.02.004

[11] DTube, "DTube Website," [accessed 2020-Apr-28]. [Online]. Available: https://d.tube/

[12] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *CoRR*, 2014. [Online]. Available: http://arxiv.org/abs/1407.3561

[13] Steem, "Steem: An incentivized, blockchain-based, public content platform," June 2018, [accessed 2020-Apr-28]. [Online]. Available: https://steem.com/steem-whitepaper.pdf

[14] SteemBlockExplorer.com, "STEEM Block Explorer," [accessed 2020-Apr-28]. [Online]. Available: https://steemblockexplorer.com/

[15] V. Bajpai, A. Brunström, A. Feldmann, W. Kellerer, A. Pras, H. Schulzrinne, G. Smaragdakis, M. Wählisch, and K. Wehrle, "The Dagstuhl Beginners Guide to Reproducibility for Experimental Networking Research," *SIGCOMM CCR*, 2019. [Online]. Available: https://doi.org/10.1145/3314212.3314217

[16] Steemit, "Trending posts - Steemit," 2019, [accessed 2020-Apr-28]. [Online]. Available: https://steemit.com/

[17] M. Thelwall, "Can social news websites pay for content and curation? The SteemIt cryptocurrency model," *J. Inf. Sci.*, 2018. [Online]. Available: https://doi.org/10.1177/0165551517748290

[18] C. Li and B. Palanisamy, "Incentivized Blockchain-based Social Media Platforms: A Case Study of *Steemit*," in *ACM WebSci*, 2019. [Online]. Available: https://doi.org/10.1145/3292522.3326041

[19] O. Ascigil, S. Reñé, M. Król, G. Pavlou, L. Zhang, T. Hasegawa, Y. Koizumi, and K. Kita, "Towards Peer-to-Peer Content Retrieval Markets: Enhancing IPFS with ICN," in *ACM ICN*, 2019. [Online]. Available: https://doi.org/10.1145/3357150.3357403

[20] L. Popa, A. Ghodsi, and I. Stoica, "HTTP as the Narrow Waist of the Future Internet," in *ACM HotNets*, 2010. [Online]. Available: http://doi.acm.org/10.1145/1868447.1868453

[21] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE Communications Surveys and Tutorials*, 2016. [Online]. Available: https://doi.org/10.1109/COMST.2015.2401424

[22] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "YoMoApp: a Tool for Analyzing QoE of YouTube HTTP Adaptive Streaming in Mobile Networks," in *IEEE EuCNC*, 2015. [Online]. Available: https://doi.org/10.1109/EuCNC.2015.7194076

[23] D. Tsilimantos, T. Karagkioules, and S. Valentin, "Classifying Flows and Buffer State for Youtube's HTTP Adaptive Streaming Service in Mobile Networks," in *ACM MMSys*, 2018. [Online]. Available: https://doi.org/10.1145/3204949.3204955

[24] N. Wehner, S. Wassermann, P. Casas, M. Seufert, and F. Wamser, "Beauty is in the Eye of the Smartphone Holder A Data Driven Analysis of YouTube Mobile QoE," in *IEEE ComSoc CNSM*, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8584953

[25] V. Bajpai, S. Ahsan, J. Schönwälder, and J. Ott, "Measuring YouTube Content Delivery over IPv6," *SIGCOMM CCR*, 2017. [Online]. Available: https://doi.org/10.1145/3155055.3155057

[26] R. K. P. Mok, V. Bajpai, A. Dhamdhere, and kc claffy, "Revealing the Load-Balancing Behavior of YouTube Traffic on Interdomain Links," in *PAM*, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-76481-8_17

[27] C. Huang, J. Li, and K. W. Ross, "Can Internet Video-on-Demand be Profitable?" in *ACM SIGCOMM*, 2007. [Online]. Available: https://doi.org/10.1145/1282380.1282396

[28] J. Wang, C. Huang, and J. Li, "On ISP-Friendly Rate Allocation for Peer-assisted VoD," in *ACM MM*, 2008. [Online]. Available: https://doi.org/10.1145/1459359.1459397

[29] B. Li, G. Y. Keung, S. Xie, F. Liu, Y. Sun, and H. Yin, "An Empirical Study of Flash Crowd Dynamics in a P2P-Based Live Video Streaming System," in *IEEE GLOBECOM*, 2008. [Online]. Available: https://doi.org/10.1109/GLOCOM.2008.ECP.339

[30] C. Wu, B. Li, and S. Zhao, "Diagnosing Network-Wide P2P Live Streaming Inefficiencies," in *IEEE INFOCOM*, 2009. [Online]. Available: https://doi.org/10.1109/INFCOM.2009.5062221

[31] Z. Li, J. Lin, M. Akodjènou, G. Xie, M. A. Kâafar, Y. Jin, and G. Peng, "Watching Videos From Everywhere: A Study of the PPTV Mobile VoD System," in *ACM IMC*, 2012. [Online]. Available: https://doi.org/10.1145/2398776.2398797

[32] S. Zou, Q. Wang, J. Ge, and Y. Tian, "Peer-Assisted Video Streaming With RTMFP Flash Player: A Measurement Study on PPTV," *IEEE Trans. Circuits Syst. Video Techn.*, 2018. [Online]. Available: https://doi.org/10.1109/TCSVT.2016.2601962

[33] A. Raman, S. Joglekar, E. D. Cristofaro, N. Sastry, and G. Tyson, "Challenges in the Decentralised Web: The Mastodon Case," in *ACM IMC*, 2019. [Online]. Available: https://doi.org/10.1145/3355369.3355572

[34] S. Ahsan, V. Singh, and J. Ott, "Impact of Duration on Active Video Testing," in *ACM NOSSDAV*, 2016. [Online]. Available: https://doi.org/10.1145/2910642.2910651

[35] Google Help Center, "YouTube Help: Manage ad breaks in long videos," [accessed 2020-Apr-28]. [Online]. Available: https://support.google.com/youtube/answer/6175006