

To Cap or not to Cap: Bandwidth Capping Effects on Network Interactions and QoE of Competing Short Video Streams

Nikolas Wehner
University of Würzburg
Würzburg, Germany
nikolas.wehner@uni-wuerzburg.de

Theo Karagioules
AT&T Labs – Research
Bedminster, NJ, USA
theo@research.att.com

Emir Halepovic
AT&T Labs – Research
Bedminster, NJ, USA
emir@research.att.com

Filip Simonovski
University of Augsburg
Augsburg, Germany
filip.simonovski@uni-a.de

Tobias Hoßfeld
University of Würzburg
Würzburg, Germany
tobias.hossfeld@uni-wuerzburg.de

Michael Seufert
University of Augsburg
Augsburg, Germany
michael.seufert@uni-a.de

ABSTRACT

Delivering popular short video streaming services like TikTok, Instagram Reels, or YouTube Shorts, poses substantial challenges for service providers and network operators. This is not only due to high download volumes but also due to high-volume pre-loading strategies that cause high bandwidth demand variations. These strategies, designed to reduce initial delays, can additionally lead to bandwidth excess when users swipe quickly through videos and consume only a fraction of downloaded content. This creates inefficiencies and unbalanced network resource utilization, particularly in competitive bandwidth environments. To address these challenges, we investigate the effectiveness of bandwidth capping in this paper, i.e., limiting the throughput of video flows in the network. We conduct measurement studies to analyze the impact of capping on network interactions and Quality of Experience (QoE) of three popular short video services in different scenarios. We find that capping substantially reduces download volume (15% – 45% median reduction) and bandwidth excess (18% – 52% mean reduction), while bandwidth utilization fairness improves. Meanwhile, QoE surprisingly remains nearly unaffected in most cases, with only minor statistical differences in a few scenarios.

CCS CONCEPTS

• **Networks** → **Network management**; **Network measurement**.

KEYWORDS

Short-form Video, QoE, Streaming, Traffic Management

ACM Reference Format:

Nikolas Wehner, Theo Karagioules, Emir Halepovic, Filip Simonovski, Tobias Hoßfeld, and Michael Seufert. 2025. To Cap or not to Cap: Bandwidth Capping Effects on Network Interactions and QoE of Competing Short Video Streams. In *The 16th ACM Multimedia Systems Conference (MMSys '25, March 31–April 4, 2025, Stellenbosch, South Africa)*.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MMSys '25, March 31–April 4, 2025, Stellenbosch, South Africa
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1467-2/25/03
<https://doi.org/10.1145/3712676.3714441>

'25), March 31–April 4, 2025, Stellenbosch, South Africa. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3712676.3714441>

1 INTRODUCTION

Short video services like TikTok, Douyin, YouTube Shorts and Instagram Reels, have become popular very quickly and gained billions of users [9]. Short videos are typically a few seconds to about a minute long and are mostly recorded on smartphones in vertical orientation, unlike long-form content, which is usually produced with professional equipment. Short video platforms generally present content to users algorithmically, based on individual preferences, their social interactions, or global content popularity. The main user interaction is swiping, i.e., selecting the next recommended video by moving the finger up or across the phone's touchscreen. Short videos generally follow HTTP adaptive streaming (HAS) principles, i.e., video quality can change within or between videos to adapt to varying network conditions as decided by the adaptive bitrate (ABR) algorithm. However, they exhibit distinct characteristics compared to long videos, particularly fewer encoding variants [42] and different QoE objectives. It is still highly important to deliver uninterrupted video stream, but due to the shorter duration of videos, users now expect immediate access to content. Thus, eliminating initial delay and delays between videos has become the key QoE objective, but may be hard to attain in mobile networks where conditions can fluctuate.

Both network operators and short video services have to cope with the challenging combination of large amounts of generated traffic and stringent QoE objectives. In addition, short video traffic exhibits highly variable demand resulting from the on-off pattern of the pre-loading strategies [42], composed of large bursts and idle periods. These variations are especially challenging on links with high competition and might negatively impact other traffic, which can lead to reduced fairness in bandwidth utilization and QoE. Finally, the high-volume pre-loading behavior of apps can also result in high bandwidth excess. This occurs if users swipe to the next video before consuming most of the pre-loaded content, which contributes to more competition with other applications. Short video services in turn observe varying bandwidth on the bottleneck links and have to adapt rapidly to the conditions they contribute to themselves.

Similar but less pronounced issues occur for long video services and have recently been tackled with a cross-layer approach by

Spang et al. [32], which showed that smoothing video traffic with application-informed pacing can improve its interactions with other Internet traffic while maintaining the same or better QoE. However, this approach requires combining a custom ABR algorithm with limiting the server’s sending rate, which is entirely within the domain of content and service providers. An alternative approach is to implement bandwidth capping only at the egress of a CDN edge node or inside the core or access network, to limit the throughput of video streams, thereby also smoothing traffic and promoting fairness at the network layer. This can be appealing, especially for network operators, as it allows to positively affect QoE independently of service providers, while saving network resources at the same time. However, there is no definitive answer yet on whether to cap or not to cap. In the best case, this might trigger a benign adaptation behavior of video services and could improve bandwidth utilization and QoE. In the worst case, QoE could be degraded by capping, which should be avoided. In this work, we investigate the impact of bandwidth capping on three popular, short video services – TikTok, Instagram Reels, and YouTube Shorts. While we focus on the impact of capping on bandwidth and QoE fairness within and between different short video services, we also analyze interactions with long video, using YouTube and Facebook Watch, and file download cross-traffic. We conduct automated measurement studies on smartphones competing for bandwidth on a bottleneck link, as typically found in mobile networks, and treat all services as black-boxes. Finally, we evaluate the impact of capping on unconsumed data (downloaded but not watched) considering different swiping behaviors. Our key insights into short video services are:

- Capping substantially reduces download volume (15% – 45% median reduction) and bandwidth excess (29% – 52% for two short video services, and 18% – 45% for one service).
- Capping improves bandwidth utilization fairness within and between short video services.
- Surprisingly, only a very small, service- and scenario-dependent penalty is observed on QoE metrics. One service showed no significant degradation in any QoE metric or capping scenario. One service was only affected in low capacity scenarios, where the worst-case initial delay exceeded 1 second and stalls increased by a factor of 3. One service consistently exhibited small noticeable reductions of visual quality by 6 – 9 VMAF points.
- Finally, capping allows short video services to be more friendly to other Internet traffic (file downloads and long-form videos), which in some cases benefit in performance.

Thus, we find that capping can be considered a reasonable traffic management tool with benefits for all stakeholders – network operators, service providers, and end users of video services.

Section 2 outlines related work. Section 3 presents the measurement methodology and experimental setup. We analyze network performance metrics in Section 4 and QoE performance in Section 5. In Sections 6 and 7, we investigate the impact of cross-traffic and swiping behavior, respectively. We discuss findings, limitations, and future work in Section 8 before Section 9 concludes.

2 RELATED WORK

Characteristics of Short Video Streaming Services. With the growing popularity of short video services, several works have been

conducted to better understand their high-level characteristics, such as distribution of video duration and bitrate [4, 40].

Video Streaming Measurements. Various tools have been proposed for measurement-based performance evaluation of video streaming applications. A web browser plugin that collects data [16, 31, 33] can be automated to collect measurements using Selenium browser automation tool [22, 26]. A custom Android app can be utilized in a local testbed for network traffic and QoE monitoring [29, 35]. However, these approaches do not use native smartphone apps directly. To overcome this limitation, a wrapper [30] was introduced to leverage Android Debug Bridge (ADB) to automatically run measurements and interact with native app’s UI elements. Zhu et al. [42] conducted automated measurement studies for short video services to examine pre-loading mechanisms and data consumption. In this work, we build upon prior approaches and present a comprehensive measurement framework for fully automated measurements, which is able to collect data both within native apps and in the network at the same time. It further allows for automated post-processing and analysis, ensuring accurate and consistent measurement results.

Quality of Experience (QoE). QoE is a widely recognized concept for assessing the subjectively perceived quality of Internet applications [3]. Seufert et al. [25] identified initial delay, stalling, and video quality as major QoE metrics for video streaming. In the case of short videos, initial delay is particularly important, as users expect instant playback when they swipe. This poses a major challenge as users’ swiping behaviors are different. To tackle this challenge, various pre-loading strategies are used, following either a conventional [18, 39, 41] or a learning-based approach [7, 21, 38]. Many strategies result in pre-loading excessive content that may not ultimately be viewed by the user due to the unique nature of swiping behavior. The probabilistic model introduced in [41] addresses this issue by determining a maximum pre-load size, resulting in a 22.8% reduction in bandwidth excess. In [21], the authors formulate a QoE-data trade-off. By leveraging information about past users’ viewing duration, they manage to reduce unconsumed data by up to 58% compared to existing methods.

QoE Fairness. The approaches presented so far are focused on a single user. However, due to the high popularity of video services, it is likely that multiple video streaming clients will share a bottleneck link [17]. Significant contributions to the understanding of how to measure and achieve fairness with respect to QoE are provided in [8, 24]. Improving fairness among competing video clients has also been a subject of research on congestion control. Minerva, a protocol introduced in [17] adjusts the bandwidth allocation of clients to maximize QoE fairness, resulting in a 24% improvement of the worst viewing experiences. Moreover, VSIM [37], a mobility-profiled QoE-driven bandwidth allocation strategy, enhances QoE fairness by 40% by using a central server that prioritizes bandwidth allocation based on users’ trajectory. In contrast, the ABR algorithm TCPAL [28] has no centralized controller but still maintains QoE fairness even under fluctuating network conditions, while FlexStream uses a more centralized approach [2].

Network management. In modern network management, an important challenge for operators is to effectively orchestrate resources to maintain high QoE for users while ensuring (QoE) fairness across the network. Wamser et al. [34] analytically and simulatively evaluate a management concept that dynamically adjusts

network bandwidth according to the current traffic mix and application requirements. Their approach enhances QoE for video streaming and web browsing, demonstrating the potential of application-aware bandwidth allocation. To further optimize resource allocation, network slicing has emerged as an effective strategy, offering the flexibility to allocate network slices for specific services. Kwak et al. [14] presented a dynamic bandwidth slicing framework that optimizes resource allocation for video services, enhancing QoE and minimizing delays. Complementing this approach, research has also focused on bandwidth capping as a method to manage resources. For instance, in [13], the network and client work together to implement bandwidth capping, which helps reduce network congestion and enhances streaming performance.

3 METHODOLOGY

3.1 Measurement Study Design

Short Video Services. The goal of this work is to evaluate the benefits of network management for bandwidth utilization and QoE of short video streaming services in competition scenarios with a bottleneck link, e.g., in WiFi access networks or mobile network cells. More precisely, we want to investigate how different popular short video services – TikTok, Instagram Reels, and YouTube Shorts – are affected by capping on the bottleneck link. We treat these services as black boxes in terms of system design, adaptation logic, encoding, and congestion control. We anonymize the services for the remainder of this paper, referencing them randomly as Service 1-3, or S1-S3. To increase the realism of the measurement, and thus the generality of the results, we do not measure each service in isolation, with a single device on a bottleneck link as done in previous works. Instead, we measure the behavior and the resulting QoE using multiple devices in competition within and between services, which allows us to observe more realistic fluctuations of network conditions for the devices and services. In our testbed, we use six devices, with two devices for each service.

To allow for reliable, consistent, and reproducible measurements, interaction with the devices is automated using Appium [1]. The experiments start by clearing app caches, followed by starting the app. Then, each app is navigated to a predefined video playlist via the automated UI, as detailed in Section 3.3. All devices wait for a randomized duration of 0 to 5 seconds before starting playback. This prevents identical video startup sequences and avoids measurement bias from varying navigation times.

Since swiping behavior greatly impacts throughput requirements, we implement three swiping modes: *deterministic*, *half*, and *full*. In deterministic mode, a swipe occurs after each video is played for 15 seconds (regardless of its duration), reflecting the shortest observed median playback duration across services in our dataset and supported by prior findings [40]. The swipes in the other two modes occur at the half or full playback points (cf. Figure 2b).

Cross-Traffic. Since network management affects both short video flows and interactions with other cross-traffic, we consider two cross-traffic scenarios in this work. For this, we add two more smartphones to the testbed, which produce cross-traffic during the short video measurements. In the first scenario, we analyze the behavior of short video services in the presence of long videos. In this scenario, the two cross-traffic devices stream long videos, in

particular, one streams YouTube and the other one Facebook Watch, representative black-box long video services. Again, we anonymize these services for the remainder of this work, referring to them as L1 and L2 in a randomized order. Each device streams a different video playlist. In the second scenario, we analyze the behavior of short video services in the presence of file downloads. Here, the two cross-traffic devices download a large ISO file from an Ubuntu server. For automating the file download, we use Selenium [23]. To start the file download, we open the Chrome browser on the end-devices and enter the download URL of an Ubuntu ISO. This ISO can never be completely downloaded during the experiment duration due to its size. File downloads are also loosely synchronized with app startups, i.e., they start around the same time as video playback.

These scenarios were selected to cover not only two other common service types, long video and file downloads, but also to include highly diverse cross-traffic characteristics. Similar to short videos, long videos have bursty traffic and short flow duration due to their typical on-off-behavior. During flow start they are fighting for their bandwidth share, subject to congestion control on transport layer (TCP/QUIC). Their network requirements are also affected by the ABR algorithms on application layer, and thus, they are able to adapt their network requirements to changing network conditions and competition. Contrary, file downloads are typically large, long-running flows. These flows exhibit converged TCP behavior, which means they aim to saturate the available capacity and will only adapt their network requirements in case of congestion as a result of TCP congestion control. Moreover, they are known to have advantages over short flows when competing for spare capacity, which can lead to a disproportionate share of capacity. By investigating these different cross-traffic characteristics, we are able to observe a wide range of the effects of capping on video streams.

Network Management. We consider three different bottleneck capacities (12, 16, or 20 Mbps) for the eight devices in our testbed, and compare two network management configurations: unconstrained and capping. In the unconstrained scenario, all video flows are assigned a low guaranteed download rate of 32 kbps to avoid starvation. All flows have a maximum allowed download rate equal to the overall capacity in the network. A fair bandwidth share between the eight devices for these capacities would thus result in 1.5 Mbps, 2.0 Mbps, and 2.5 Mbps for each device, respectively. In the capping configuration, we again guarantee all video flows a download rate of at least 32 kbps to avoid starvation, but cap the maximum allowed download rate to the fair shares of 1.5, 2.0, or 2.5 Mbps, respectively. The total network capacity equals eight times the capped download rates (12, 16, or 20 Mbps), making it comparable to the unconstrained scenario.

Performance Metrics. We evaluate the effect of bandwidth capping from both a network and a QoE performance perspective. In terms of network performance, we evaluate the peak throughput of the flows and the overall download volume per experiment. Additionally, we analyze the bandwidth utilization of each device, which we define as the fraction of traffic that was downloaded by this device compared to the theoretical maximum, i.e., the traffic that could be downloaded when fully utilizing the available capacity. To evaluate the QoE of the short video services, we focus on QoE factors that are widely accepted in literature [25], and by well-established video QoE models, e.g., the standardized video QoE model ITU-T

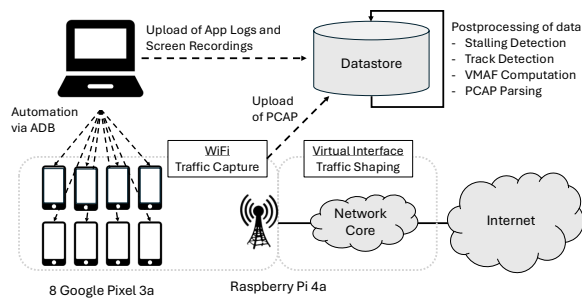


Figure 1: Simplified testbed setup.

P.1203 [10]. We assess the visual quality of the streamed videos by applying the Video Multimethod Assessment Fusion (VMAF) [15] method. It compares the quality of the streamed video to an undistorted reference video with a VMAF score of 100. The more the visual quality of the streamed video is reduced, the lower the resulting VMAF score. The smallest unit which can be perceived by a user – Just Noticeable Difference (JND) – is a score difference of ± 6 [20] for VMAF, with the implication that the larger the absolute VMAF difference, the more noticeable a quality switch would be. Apart from visual quality, the startup delay or initial delay, i.e., the delay before the video starts its playback, is another important QoE factor, especially for short video services, where users expect instantaneous playback start after swiping. Finally, we evaluate stalling, i.e., playback interruption due to rebuffering, which is considered the most important video streaming QoE factor [25, 27]. We present the total stalling duration here, representing the total waiting time during a single experiment run.

3.2 Measurement Setup

Figure 1 shows an overview of the testbed setup, which includes a PC, a Raspberry Pi 4a, eight Google Pixel 3a smartphones (Android 11), and an external server as a central data store. The PC automates app interactions on smartphones using Appium or Selenium with Android Debug Bridge (ADB) [5] and uploads logs and screen recordings to the datastore. The Raspberry Pi’s WiFi interface acts as an access point for the smartphones, captures the network traffic in a PCAP file, and uploads it to the datastore. The Raspberry Pi’s Ethernet interface routes all traffic to the Internet, allowing normal video service use. A virtual interface on the Raspberry Pi between these interfaces emulates the network core and is responsible for capping. Once the data is uploaded after the measurements, automated analysis identifies stalling events and ABR tracks from the screen recordings and computes VMAF for each video. To achieve this, we created a measurement framework featuring a dockerized core and various nodes that handle tasks such as Appium automation and screen recording post-processing. We use Apache Kafka and the publish-subscribe model for core-node communication.

Application Layer. In our testbed, all eight smartphones are connected via USB hub to the PC. Eight service nodes run on this PC, each with its own Appium or Selenium process for interacting with Android apps or performing file downloads on the respective devices. Each process distinguishes between the devices by using

unique Android device IDs and ports, and then executes an automation script tailored for the configured service. We utilize the screen recording capabilities of Appium to record the video playbacks. During automation, we log timestamps of app interactions, e.g., swipes, and video IDs for each device individually. These logs help synchronize application and network data during analysis.

To coordinate the timings of the eight devices and ensure controlled competition, we use a synchronization node. This node synchronizes the service nodes by splitting the automation script into steps, i.e., app cache clearing, startup, and playback. Devices wait for others to complete each step before being signaled to proceed.

Network Layer. On the network side, we use one node for capping traffic with Linux `tc` and another for traffic capture with `tcpdump`, both on the WiFi interface of the Raspberry Pi 4a. Each smartphone is assigned a static IP address to facilitate capping. We use Hierarchical Token Bucket (HTB) for classful queuing to cap downlink traffic across different IP address ranges for each service. In detail, we define one class for the competing downlink video traffic and one class for all other non-competing traffic. Inheriting from the competing class, each device and video service creates another child class for capping device and video flows. We are able to exclude non-video traffic by marking all traffic flows belonging to specified service IP address ranges collected in advance.

Track Detection. The track detection node identifies the streamed quality levels over time to estimate QoE change with traffic capping. For YouTube, we monitor the track directly from the app itself using the built-in StatsForNerds option [12, 30]. Here, we immediately activate the StatsForNerds overlay after starting video playback. The overlay shows the video track, buffered video data, and more, and includes a copy button to log this information. By copying and logging this data periodically throughout video playback, we collect the relevant track information.

For all other services, we must rely on alternative means as they do not expose such statistics to the public. For this purpose, we utilize and adapt VideoEye [36], which requires to download all tracks of the original video from the video service platform as a reference. Then, VideoEye temporally aligns all screen recording frames with all frames of the reference video based on distances between downscaled, low-resolution versions of the frames. Finally, VideoEye computes the distances between screen recorded frames and all aligned reference frames for all reference tracks. At each point in time, the detected track of the screen recording is set to the reference track with the smallest distance.

In the original VideoEye evaluations, the authors had full control of the video sources and video player. In this work, we had to adapt VideoEye to make it compatible with all measured video services. First, the screen recorded frames and the reference frames must match pixel-wise. As the viewports of the apps usually do not fill the entire screen, we have to determine not only the measures of the viewport, but sometimes also how much of the source video is cut off due to viewport scaling. This requires exactly aligned cropping and scaling of screen recorded video and reference videos. Next, short video services usually overlay non-static text and icons on top of the actual video, e.g., the number of likes. To prevent these elements from skewing the error between screen recorded and reference frames, we mask them in both videos. Without masking, the pixel-wise error would be generally larger and could hide subtle

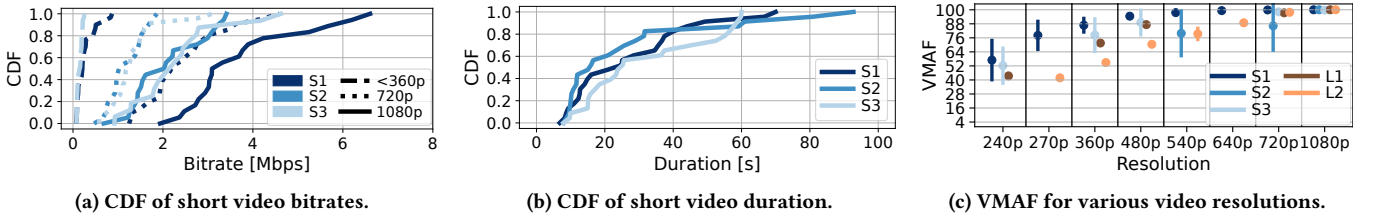


Figure 2: Statistical description of predefined short video playlists and long videos.

Table 1: Overview of conducted experiments. P1 and P2 correspond to playlists 1 and 2. The baseline scenario is marked in blue.

Traffic Management	Max. Bandwidth Per Device	Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	CT Scenario	Device 7	Device 8	Swiping
Unconstrained	12/16/20 Mbps	S1 (P1)	S1 (P2)	S2 (P1)	S2 (P2)	S3 (P1)	S3 (P2)	Long Video	L1 (P1)	L2 (P1)	Deterministic
											Half
											Full
								FD	FD (Unc.)	FD (Unc.)	Deterministic
Capped	1.5/2.0/2.5 Mbps	S1 (P1)	S1 (P2)	S2 (P1)	S2 (P2)	S3 (P1)	S3 (P2)	Long Video	L1 (P1)	L2 (P1)	Deterministic
											Half
											Full
								FD	FD (Unc.)	FD (Unc.)	Deterministic

track differences. Finally, video players manipulate the color space of the videos by changing contrast and brightness, resulting in slightly different colors between screen recorded and reference videos. To address this, we use histogram matching to align the color space of the reference videos with the screen recorded video.

Stalling Detection. To identify stalling events during playback, we utilize the screen recordings and FFmpeg. FFmpeg provides a filter called freeze-detect, which detects freezes by comparing differences between the components of the video frames. Independent of the service, we use the default noise tolerance (-60dB) and a custom duration of 0.5 seconds required to identify a set of frames as frozen. We mask all visuals not linked to the actual video, e.g., progress or loading bars, to avoid interference during detection. Finally, we post-process the detected stalling events by aggregating stalling events occurring rapidly after each other to a single event and by adding a single long stalling event in case the video never stopped freezing after startup.

3.3 Experimental Design

Ideally, we would emulate typical short video app usage in our experiments: starting the app, selecting short videos if needed, and watching a sequence of recommended videos. However, pure recommendation-based streaming introduces randomness, which hinders reproducibility between iterations due to the differences between random videos. Thus, we generate a pre-defined playlist for each device. This ensures that each device consistently uses the same app and streams the same videos in every run, though playlists and videos vary between devices.

Playlist Characteristics. In the following, we briefly characterize the selections for the short video playlists and long videos with respect to bitrates, video duration, and VMAF in Figure 2. For each short video service, we select 24 videos with varying contents including high-motion (sports), medium-motion (dance clips and concerts), and low-motion videos (slowly wandering camera across

a landscape), all with the aspect ratio of 9:16. We divide these videos into two playlists containing 12 videos each. We balance the network requirements of contents between the playlists per service and shuffle the contents only within the playlists.

The distribution of video bitrates for three selected resolutions, i.e., quality levels, of the video playlists are depicted in Figure 2a as CDF, where the x-axis denotes the video bitrate in Mbps. It shows that the bitrates range from 66 kbps to 6.6 Mbps and that S1 generally requires higher bitrates and network resources than the other services. Figure 2b shows the distribution of the video duration across the three services. The video duration ranges from around 6.8 seconds to around 92.8 seconds, while the median is 22.5 seconds. Generally, the video duration is comparable across the services. Finally, Figure 2c illustrates VMAF scores for various resolutions compared to a 1080p reference video. The markers show the average VMAF score across all measured videos of a service and the error bars indicate the standard deviation. We observe that the different services use different representation sets, and thus, different resolutions. For example, S3 offers only resolutions of 540p and above. We also find that, due to different encoding and compression settings, the visual quality of the resolutions are highly variable between services. For example, VMAF scores for S1 are already close to 100 for 540p, while for L1 they approach 100 only from 720p. We omit existing resolutions below 240p for clarity here.

Experiment Summary. We provide an overview of all conducted experiments in Table 1. Note that the maximum available bandwidth per device is always only one of the provided options, leading to 24 different experiment settings. Devices 7 and 8 always remain unconstrained in the file download (FD) cross-traffic scenario, even if the short video services are capped. From June to August 2024, we performed 50 consecutive runs for each of the 24 experiments, resulting in 1200 measurements in total. After filtering invalid or failed measurements, 1001 measurements remain with a mean of 37.07 and a minimum of 29 runs per experiment setting.

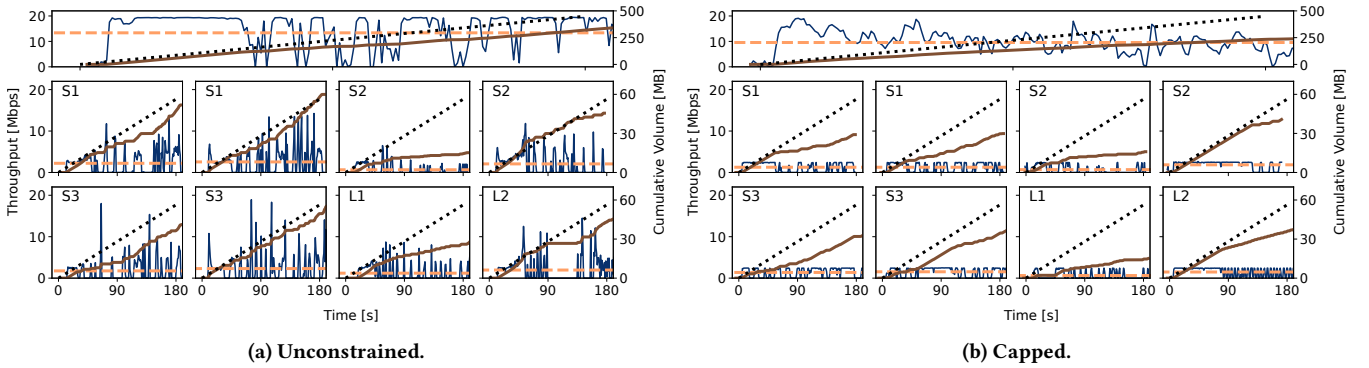


Figure 3: Visualization of network performance metrics for two exemplary measurement runs. Dark blue lines denote throughput. Brown lines denote cumulative download volume. Dashed orange line represents average throughput over measurement time. Dotted lines indicate maximum achievable download volume overall and per-device fair share.

4 NETWORK PERFORMANCE

In this section, we analyze network traffic and the associated performance metrics. To illustrate the richness of insights we obtained from our experiments, Figures 3a and 3b present the results of two exemplary measurements for a bottleneck capacity of 20 Mbps in the unconstrained and capped conditions, respectively. Both runs face long video cross-traffic and deterministic swiping. The wide plot in the first row depicts overall traffic on the bottleneck link, while the second and third rows display per-device traffic. The plots show the measurement time in seconds [s] on the x-axis. The dark blue lines represent throughput in Megabits per second [Mbps], the brown lines indicate cumulative download volume in Megabytes [MB], and the orange lines show the average throughput over the measurement period, all displayed on the y-axis. Additionally, the dotted lines denote the maximum achievable download volume over time, both in regard to the overall link capacity of 20 Mbps (top row) and to the fair share per device of 2.5 Mbps (bottom rows).

The overall traffic in the unconstrained network (top of Figure 3a) shows that the average throughput over time was 13.37 Mbps (dashed orange line), which corresponds to an average bandwidth utilization of 66.8%. Initially, bandwidth competition peaks as services try to fill their buffers with the video currently being viewed and pre-loaded videos. Thus, we observe a period of peak download throughput (dark blue curve), in line with the expected initial buffering behavior of video streaming. This is followed by typical on-off traffic patterns of the devices, which in turn result in several utilization drops on the aggregate registered approximately at the 45 s mark and lasting until the end of the experiment. At this point, the slope of the cumulative download volume (brown curve) slightly decreases but maintains a linear trend until the end.

Looking at the individual devices in the bottom rows, we see that initially, all devices download around their fair share, although the second set of devices (right-most) of S1-S3 all show a large throughput spike above the fair share in the first seconds, an adverse effect of unconstrained competition. These devices also keep their bandwidth utilization around the fair share for the rest of the measurement run. For the other devices, i.e., the first (left-most) devices of S1-S3, the brown curve becomes much lower than the dotted black line indicating a reduction in their network requirements,

which results in lower fairness in terms of bandwidth distribution. This behavior could be attributed to the ABR algorithm employed by the services, i.e., a reduction of the requested video bitrate. This suggests that their visual quality, and thus, their QoE will be lower, showcasing the unfavorable effect of competition on these three devices. Considering the cross-traffic, which is long video, we do not observe a large negative impact due to competition for this measurement run. After the initial burst phases, where both devices receive around their fair share, L1 shows the expected on-off pattern with throughput spikes every few seconds. In comparison, L2 has a longer initial burst phase up to around 90 seconds, followed by a 45-second pause, before initiating a second burst phase. Accounting for potential differences in system design between the two services, both behaviors can be considered normal.

In the capped scenario, shown in Figure 3b, the overall traffic reveals an average throughput of 9.56 Mbps, which corresponds to an average bandwidth utilization of 47.8%, substantially lower than in the unconstrained scenario. More specifically, compared to the unconstrained scenario (349 MB), the total downloaded volume was significantly lower, with a final value of 245 MB. Unlike the unconstrained scenario, the throughput (dark blue curve) never saturated the link capacity but remained mostly close to the average. This demonstrates that the network load and the overall burstiness of traffic can be effectively reduced by capping. Looking at the devices individually, we can observe that while the characteristic on-off behavior still exists, we encounter more extended on-phases, and sometimes even continuous, such as in the second device of S2. Both devices of S2 show a similar cumulative download volume (brown curve) compared to the corresponding unconstrained scenario in Figure 3a. S1 and S3, on the other hand, download substantially less compared to the unconstrained scenario, which depending on codecs and the effect of bitrate adaptation might also result in a reduced visual quality, and thus, a reduced QoE. Similar observations can be made for the cross-traffic, where L2 downloads around the same, but L1 downloads substantially less compared to the unconstrained scenario. With network management, fairness increases across devices and services when compared to unconstrained network utilization. Nonetheless, as a static network management approach, capping may affect QoE differently depending on the service's design, i.e., adaptation and pre-loading algorithms.

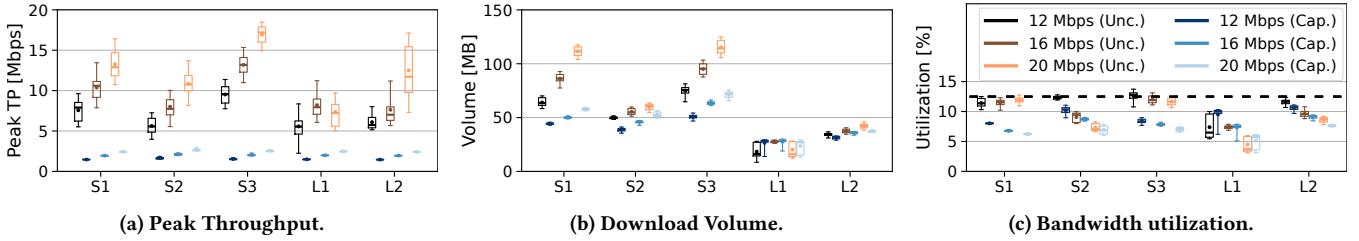


Figure 4: Comparison of network performance metrics.

Having shown exemplary measurement runs, we now focus on aggregated results for the remainder of this work. Figure 4 compares the distribution of network performance metrics, namely, peak throughput, download volume, and utilization. For each service and bandwidth setting, we display the observed distributions as box plots for both unconstrained (brown) and capped (blue) scenarios side by side. The different maximum download rates are indicated by color brightness: dark for 12 Mbps, medium for 16 Mbps, and light for 20 Mbps. Boxes span from the 25th to 75th percentile with thick lines indicating the median (50th percentile) and circles indicating the mean, while whiskers extend from the 5th to 95th percentile. The x-axis displays services and the y-axis corresponding metrics.

First, we observe that the peak throughput behaves as expected (cf. Figure 4a): in the unconstrained scenario, the throughput can theoretically reach up to the link capacity, though it never actually does due to competition. Naturally in the capped scenario, the peak throughput of all services is limited by the bandwidth cap.

Figure 4b displays the distribution for downloaded volume. For the short video S1-S3, it is evident that capping reduces the overall downloaded volume. While S1 and S3 generate from 30% to 45% less median volume compared to the unconstrained scenario, the volume difference is less pronounced for S2, where the median decrease is between 15% and 20%. We also observe that under capping, all short video services download similar amounts of data, thus achieving higher network fairness between services. Another important observation is that capping substantially reduces the variation of download volume of all services as indicated by the shorter boxes in the capping scenario. This shows that capping also leads to higher network fairness within services. While lower data volume also signals more efficient network usage and benefits to data-constrained users and rate plans, we will examine the impact on QoE when we look at Figure 5 in Section 5. For the long video L2, capping does not reduce the download volume much but again variation decreases. However, we can make an interesting observation for L1. Here, we see from the increased median in the darkest (12 Mbps) and lightest (20 Mbps) boxes that the download volume increases in the capping scenario compared to the unconstrained scenario. Thus, we find that capping has the potential to make short video flows more friendly neighbors, aligning with prior independent results in [32]. However, our results also indicate that the effect of capping may depend, aside from the design of the individual video applications as we established previously, on the type of video traffic as well. Since short and long video traffic may behave differently in capped scenarios, traffic management techniques tailored to video types could be more efficient and preferable.

The box plots for bandwidth utilization in Figure 4c reveal lower fairness in the unconstrained scenario even though devices rarely exceed the fair share utilization at 12.5%, as indicated by the dashed line depicting the results of the capping scenario. We see large differences between services and also a high variation in utilization within services. As intended, with capping, the fairness is increased especially within services, as indicated by the shorter boxes. There are still fairness imbalances between services but to a lesser extent.

5 QOE PERFORMANCE

Network performance results revealed an increase in network-level fairness with capping but also a decrease in bandwidth utilization, potentially degrading application performance and QoE. Since QoE is crucial for end users, we will now analyze the resulting impact of capping on QoE for our baseline scenario. To assess the impact of capping on the application layer, we analyzed VMAF, initial delay, and total stalling duration, with results presented in Figure 5. Again, the results for the unconstrained scenario and the capping scenario are shown side by side (similarly to those in Figure 4).

VMAF results (cf. Figure 5a) show that capping had minimal impact on the visual quality of S1 and S2 videos, while S3 experienced some decreases in VMAF. Note that the y-axis starts at 76 to better highlight the differences. The average VMAF decreases of S2 are between 6 and 9, and thus, above the JND threshold of 6 VMAF points [20]. This means that end users are expected to notice the reduced visual quality, which could negatively affect the QoE for this service. In the case of long video services, we see again that L1 benefits in VMAF values for 12 Mbps capacity (dark colors), while the VMAF scores with other capacities are mostly unaffected, which is in line with network performance metrics. For L2, we note that under the lowest capacity conditions, VMAF slightly decreases. Under higher capacities, we note similar VMAF scores and a less pronounced effect of capping. Further, the results are concentrated in a tight box (meaning higher fairness) around the median.

In summary, visual quality is not affected negatively in most capping scenarios, for all video services. Only S3 exhibits a consistent, but small reduction of visual quality, and potentially, overall QoE. To put these QoE results into perspective with the corresponding bandwidth utilization on the network layer, we analyze the trade-offs between cumulative download volume and VMAF scores for the different services in Figure 6. The x-axis represents the cumulative download volume, while the y-axis illustrates the corresponding VMAF scores. The averages in the unconstrained and capped scenarios are depicted with a circle and a cross, respectively. Horizontal and vertical lines show the standard deviations, solid

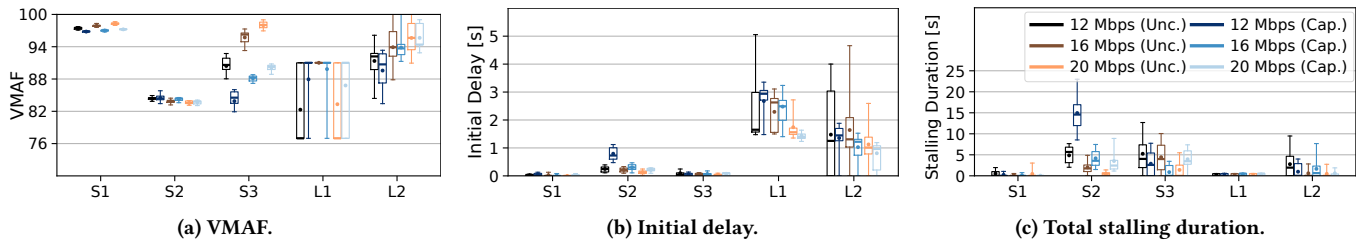


Figure 5: Comparison of QoE performance metrics.

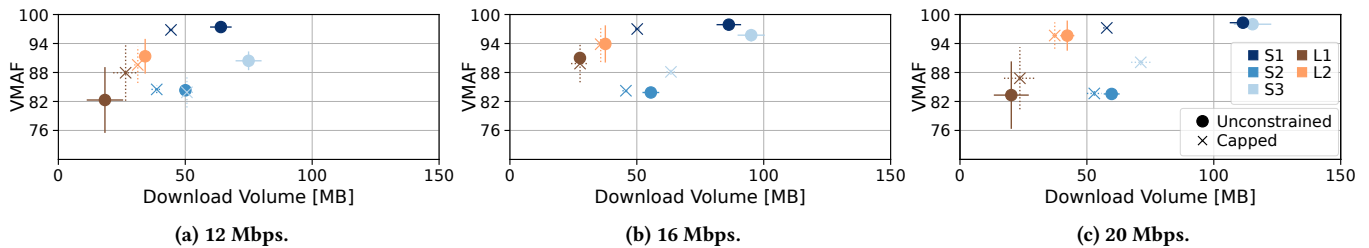


Figure 6: Trade-off between VMAF and download volume for each video service.

for unconstrained and dotted for capped. The analysis is conducted separately for the three available bottleneck capacities.

We can clearly see three behaviors with respect to the trade-off between VMAF and network utilization (download volume) as capping is applied: (1) substantial reduction of utilization without noticeable impact on visual quality, (2) small or insignificant change in both VMAF and utilization and (3) significant reduction in both VMAF and utilization. The first behavior resembles over-utilization of bandwidth without visual benefit prior to capping, which represents the ideal and best case of applying capping. In the figures, it can be observed as a horizontal shift leftward between unconstrained and capped values for S1. The second behavior may represent services that are already conservative in both network utilization and visual quality. Exemplars are S2 from the short video side, but also long video L1 and L2. This behavior appears as a minimal shift in any direction in the figures. Finally, the third behavior resembles maximized utilization and visual quality that are both sensitive to changes in network conditions. This is exemplified by S3 and appears as a diagonal shift towards lower values from unconstrained to capped points across both axes. This behavior is the least amenable to capping compared to prior two.

In summary, as four out of five services exhibit benign behaviors, we see mostly beneficial trade-offs with capping. However, this analysis explains only the network efficiency profile of services but must be combined with initial delay and stalling results for a comprehensive understanding.

Initial delay is especially important for short video streaming, as users expect no waiting time after swiping to the next video. In Figure 5b, we can generally see that all short video services (S1-S3) have much lower initial delays compared to the long video services (L1-L2). The initial delays of S1 and S3 are always below 100 ms, which is the limit for having the user feel that the system is reacting instantaneously [19], and they are not negatively affected by capping. Only S2 has an increase in initial delays with capping.

Higher delay is especially prominent under the lowest bottleneck capacity, where startup times might exceed 1 second, which may be considered too long [40]. These delays should be clearly noticeable by users and could interrupt the users' flow of thoughts [19], and might thus negatively impact the QoE. For L1 and L2, we see a reduction of initial delay variation with capping. Additionally, for the lowest bottleneck capacity of 12 Mbps, we have an increase in initial delays, especially for L1, for which the median increases by more than 1 second. For the other capacity limits, the median values remain around the same with capping.

As with long videos, interruptions of the playback, also called stalling or rebuffering, are considered a major QoE factor for short-form videos. Figure 5c shows the analysis of the total stalling duration, i.e., the sum of durations of all stalling events within a video session. We observe almost no stalling for S1, even with capping, while S2 faces the most significant increase in total stalling duration, especially for the lowest bottleneck capacity where the average total stalling duration almost triples. For the long videos, however, we see that L1 experiences almost no stalling and remains unaffected by capping. L2 benefits from capping under the lowest and highest capacities, but has more stalling under medium capacity.

To sum up our findings on initial delay and stalling, we see again that services S1 and S3 adjust well to capping and can perform equally well as in unconstrained scenarios. Only S2 has statistically significant increased waiting times, in the worst case noticeable initial delay above 1 second and stalling increase up to a factor of 3, which could negatively affect the QoE. While long video services see a slight degradation of initial delay and stalling in some cases, they generally appear to perform well under capping. Overall, the results show that while it is possible to design short video services to maintain QoE under capping, there is still room for improvement.

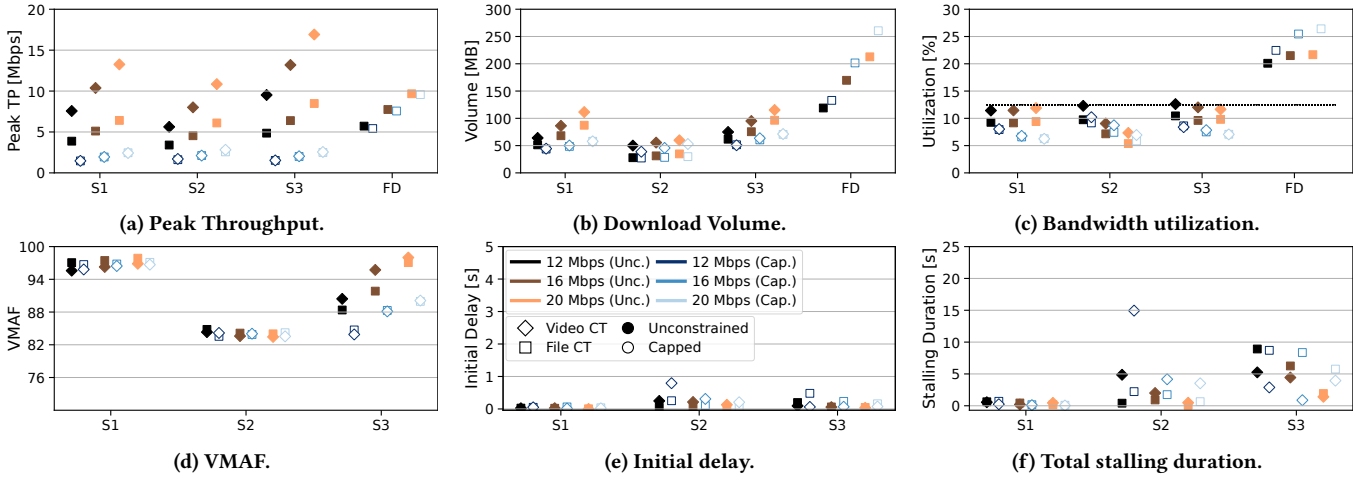


Figure 7: Cross-traffic impact on network and QoE performance metrics.

6 IMPACT OF CROSS-TRAFFIC

This section investigates the impact of capping on short video services under different cross-traffic conditions, including long video and file download traffic. File downloads remain unconstrained in the capping scenario, as we focus on the highly variable short video services. Figure 7 shows the performance of short video services (S1-S3) under different cross-traffic and capping scenarios, with respect to the different network-level and QoE metrics, in the deterministic swiping scenario. We also analyze the network performance changes for file downloads. Each marker in the figure represents the mean observed value, shaped by the cross-traffic type (diamond for long video and square for file download) and colored based on available capacity and traffic management scenario. Brown colors and filled markers correspond to the unconstrained scenario and blue colors and unfilled markers correspond to capping.

First, we observe that cross-traffic can impact the behavior of short video services at the network layer. While the on-off traffic pattern of long video allows short video services to utilize idle periods for faster data download, file download *throttles* short video services by continuously downloading in the background, leading to a much higher bandwidth utilization than long video (cf. Figure 4c and 7c). With file download cross-traffic, S1 and S3 exhibit the lower average peak throughput of up to 9 Mbps, while S2 up to only 6 Mbps in the unconstrained scenario. This is substantially lower compared to long video cross-traffic. When capping the available throughput per service, however, we observe that the differences between long video and file download cross-traffic scenarios can be reduced (S2) or entirely eliminated (S1 and S3). This indicates that capping allows more uniform, and thus, predictable network behavior from short video services irrespective of the characteristics of the cross-traffic. We note that with capping, the unconstrained file downloads exploit the remaining available bandwidth, leading to an increase in bandwidth utilization of 3 to 5 percentage points and higher download volumes with similar peak throughputs.

As we evaluate the application layer, we find that VMAF does not seem to be affected by cross-traffic, but only by traffic management. For the initial delays and the total stalling duration, we find that S1

is neither affected by cross-traffic nor by traffic management, S2 performs better in the file download cross-traffic scenario, while S3 performs better in the long video cross-traffic scenario, independent of traffic management. This behavior reflects differences in service design. While simple fixed rate limitation benefits some services and some scenarios, it might not be the best solution for other cases. Thus, studying how to improve management is required. In bandwidth competition scenarios, especially with cross-traffic, QoE could benefit from network management approaches that go beyond traffic-type agnostic capping. This includes using context information on traffic characteristics, considering the traffic mix, or accounting for differences in applications designs. Such cross-layer information exchange or joint network and service management would, however, require a framework to facilitate cooperation between network operators and video service providers.

7 IMPACT OF SWIPING BEHAVIOR

Lastly, we investigate how unconsumed data varies across different swiping scenarios. We define unconsumed data as the video data downloaded but not watched during a session. It is computed by subtracting the required bytes for watched videos from the total downloaded video bytes, using track information and bitrate metadata. The required video bytes are estimated using the identified track and video metadata, particularly the bitrate, where we assume uniform bitrate distribution across a video asset. Thus, the consumption computation is an approximation. Figure 8 depicts the mean observed unconsumed data for the short video services across different swiping and traffic management scenarios. Bars are colored by service, with filled bars representing the unconstrained scenario and hatched bars representing capping. The figure reveals that capping reduces the amount of unconsumed data across all swiping scenarios. The magnitude of reduction varies by service and available bandwidth, e.g., S1 and S3 benefit strongly from capping, while S2 benefits less. Figure 8a shows the unconsumed data savings when swiping occurs at deterministic intervals of 15 seconds (cf. [40]). Here, capping reduces the mean observed unconsumed data of S1 and S3 by 29% to 52%, and of S2 by 18% to 45%.

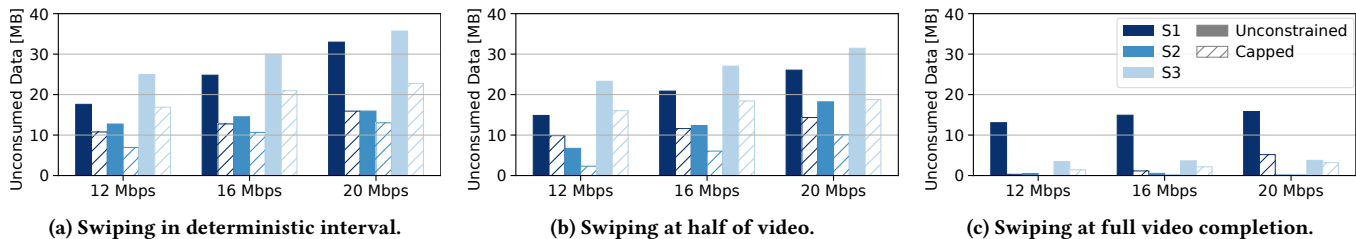


Figure 8: Impact of swiping behavior on data overconsumption.

Smaller but noticeable reductions can be observed when swiping at half of the video. The savings are smaller with longer swiping intervals compared to the deterministic condition, as a larger portion of the downloaded video is watched. When swiping at full video playback, unconsumed data should be close to zero for all services, regardless of the traffic management scenario. While this is true for S2, S1 and S3 show non-zero unconsumed data, especially in the unconstrained scenario. This is caused by communication overheads and potential inaccuracies introduced by our methodology.

Our findings in this section indicate that capping can effectively reduce unconsumed data. These savings are higher if the swiping behavior is more frequent. Generally, unconsumed data reductions have positive effects on all stakeholders. They not only facilitate reduced network utilization, but reduce CDN/server load and related costs for video service providers. Finally, it is also highly valuable for end users with limited data plans.

8 DISCUSSION

In this study, we observe that capping increases network-level fairness between different services. Generally, the data volume and unconsumed data of short video services are both significantly reduced by capping, indicating benefits to all members of the ecosystem. For network operators, capping presents an opportunity for more efficient operations, while for content providers it can yield CDN savings. Above all, end users can rely on capping as a mechanism to preserve their data plan budgets, while also ensuring that network resources are not busy over-delivering video, but instead available for all their other needs. Comparing the effect of capping across different types of video traffic, we find that the performance of short video services becomes more robust against long video or file download cross-traffic, and we observe that capping reduces the variation in network load and the application layer traffic. This leads to generally more predictable behavior, and thus, renders managing a network simpler. In addition, network utilization fairness increases across all scenarios studied with a bandwidth cap. Nonetheless, a fixed rate capping, while being a simple network management approach, may yield a small penalty in QoE, albeit variable, according to each service’s design, i.e., their adaptation and pre-loading algorithms. Generally, in terms of network performance, capping agrees with similar bandwidth regulation approaches, both directionally and quantitatively [13, 32]. For instance, the method proposed by Spang et al. [32] enhances network performance by reducing chunk throughput by up to 61% while improving QoE. Similarly, we found that capping achieves volume savings up to 45%. Furthermore, from a system design standpoint, capping is considered simpler, being

decoupled from management at the application layer. Finally, most aspects of such a system have already been implemented in the past by network operators [6, 11].

As established throughout this study, our results indicate that bandwidth capping, i.e., simple fixed rate limitation, yields clear benefits for most services and scenarios studied. Nonetheless, we also observed a very small, service- and scenario-dependent penalty in QoE for end users. Our results suggest that optimizing QoE further would require network management techniques enhanced with context awareness of the video traffic type (short-form vs. long-form) and of each application’s network requirements based on its system design. Applications, in turn, could incorporate awareness of capping techniques deployed at the network level. Thus, with respect to our future research goals, we aim to extend static rate caps toward dynamic rate caps or collaborative video traffic optimization, where network operators and video services synergize to optimize video traffic jointly. To do so, we envision that information regarding the traffic type and its QoE targets is exchanged, along with network resource availability and traffic profile requirements. Such cross-layer information exchange or joint network and service management would, however, require first the design and development of a framework to facilitate such cooperation between network operators and video service providers.

9 CONCLUSION

This work investigates whether capping of video flows, i.e., limiting their maximum network throughput, is beneficial from the perspectives of network operators, service providers, and end users. We perform a measurement study with real short video services, where six Android devices compete for bandwidth on a bottleneck link with two Android devices facilitating long video and file download cross-traffic scenarios. We analyze the effectiveness of capping on network utilization, bandwidth excess, and QoE. Our results indicate that capping is reasonable and beneficial for all stakeholders, as data volume and unconsumed data are reduced. Additionally, the fairness in the network improves. While some short video services maintain QoE, some sustain small QoE reduction under capping, meaning that there is room for improvement, but that improvement is not difficult to achieve. Overall, we find that the network becomes more predictable with capping, and thus, easier to manage. Consequently, the question should no longer be whether to cap or not to cap, as initially stated, but how to appropriately cap to maximize benefits, which we will explore as future work.

REFERENCES

- [1] Appium. 2024. Appium Documentation. <http://appium.io/docs/en/latest/> accessed 2024-09-24.
- [2] Ibrahim Ben Mustafa, Tamer Nadeem, and Emir Halepovic. 2018. FlexStream: Towards Flexible Adaptive Video Streaming on End Devices using Extreme SDN. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) (MM '18). Association for Computing Machinery, New York, NY, USA, 555–563. <https://doi.org/10.1145/3240508.3240676>
- [3] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastian Egger, Marie-Neige Garcia, Tobias Hossfeld, Satu Jumisko-Pyykkö, Christian Keimel, Mohamed-Chaker Larabi, et al. 2013. Qualinet White Paper on Definitions of Quality of Experience. (2013).
- [4] Zhuang Chen, Qian He, Zhifei Mao, Hwei-Ming Chung, and Sabita Maharjan. 2019. A Study on the Characteristics of Douyin Short Videos and Implications for Edge Caching. In *Proceedings of the ACM Turing Celebration Conference-China*. 1–6.
- [5] Android Developers. 2024. Android Debug Bridge (ADB) Documentation. <https://developer.android.com/tools/adb> accessed 2024-09-24.
- [6] Jeremy Gillula. 2016. EFF Confirms: T-Mobile's Binge On Optimization is Just Throttling, Applies Indiscriminately to All Video. <https://www.eff.org/deeplinks/2016/01/eff-confirms-t-mobiles-bingeon-optimization-just-throttling-applies> accessed 2024-09-16.
- [7] Jianchao He, Miao Hu, Yipeng Zhou, and Di Wu. 2020. LiveClip: Towards Intelligent Mobile Short-form Video Streaming with Deep Reinforcement Learning. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. 54–59.
- [8] Tobias Hossfeld, Lea Skorin-Kapov, Poul E Heegaard, and Martin Varela. 2016. Definition of QoE Fairness in Shared Systems. *IEEE Communications Letters* 21, 1 (2016), 184–187.
- [9] Mansoor Iqbal. 2024. TikTok Revenue and Usage Statistics (2024) - Business of Apps. <https://www.businessofapps.com/data/tik-tok-statistics/> accessed 2024-09-16.
- [10] ITU-T. [n. d.]. Recommendation P.1203: Parametric Bitstream-based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services over Reliable Transport.
- [11] Arash Molavi Kakhki, Fangfan Li, David Choffnes, Ethan Katz-Bassett, and Alan Mislove. 2016. BingeOn Under the Microscope: Understanding T-Mobile's Zero-Rating Implementation. In *Proceedings of the 2016 Workshop on QoE-based Analysis and Management of Data Communication Networks*. 43–48.
- [12] Theodoros Karagioules, Dimitrios Tsilimantou, Stefan Valentin, Florian Wamser, Bernd Zeidler, Michael Seufert, Frank Loh, and Phuoc Tran-Gia. 2018. A Public Dataset for YouTube's Mobile Streaming Client. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–6.
- [13] Vengatanathan Krishnamoorthi, Niklas Carlsson, and Emir Halepovic. 2018. Slow But Steady: Cap-based Client-network Interaction for Improved Streaming Experience. In *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.
- [14] Jeongho Kwak, Joonyoung Moon, Hyang-Won Lee, and Long Bao Le. 2017. Dynamic Network Slicing and Resource Allocation for Heterogeneous Wireless Services. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 1–5.
- [15] Zhi Li, Anne Aaron, Ioannis Katsavounidis, Anush Moorthy, and Megha Manohara. 2016. Toward a Practical Perceptual Video Quality Metric. *The Netflix Tech Blog* 6, 2 (2016).
- [16] Ricky KP Mok, Edmond WW Chan, and Rocky KC Chang. 2011. Measuring the Quality of Experience of HTTP Video Streaming. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*. IEEE, 485–492.
- [17] Vikram Nathan, Vibhaalakshmi Sivaraman, Ravichandra Addanki, Mehrdad Khani, Prateesh Goyal, and Mohammad Alizadeh. 2019. End-to-end Transport for Video QoE Fairness. In *Proceedings of the ACM Special Interest Group on Data Communication*. 408–423.
- [18] Duc Nguyen, Phong Nguyen, Vu Long, Truong Thu Huong, and Pham Ngoc Nam. 2022. Network-aware Prefetching Method for Short-form Video Streaming. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–5.
- [19] Jakob Nielsen. 1994. *Usability Engineering*. Morgan Kaufmann.
- [20] Jan Ozer. 2017. Finding the Just Noticeable Difference with Netflix VMAF. *Streaming Learning Center* (2017).
- [21] Nguyen Tien Phong, Truong Thu Huong, Pham Ngoc Nam, Truong Cong Thang, and Duc Nguyen. 2023. Joint Preloading and Bitrate Adaptation for Short Video Streaming. *IEEE Access* (2023).
- [22] Anika Schwind, Michael Seufert, Özgü Alay, Pedro Casas, Phuoc Tran-Gia, and Florian Wamser. 2017. Concept and Implementation of Video QoE Measurements in a Mobile Broadband Testbed. In *2017 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–6.
- [23] Selenium. 2024. Selenium Documentation. <https://www.selenium.dev/> accessed 2024-09-24.
- [24] Michael Seufert. 2021. Statistical Methods and Models based on Quality of Experience Distributions. *Quality and User Experience* 6, 1 (2021), 3.
- [25] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zimmer, Tobias Hossfeld, and Phuoc Tran-Gia. 2014. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys & Tutorials* 17, 1 (2014), 469–492.
- [26] Michael Seufert, Raimund Schatz, Nikolas Wehner, and Pedro Casas. 2019. Quicker or Not? - An Empirical Analysis of QUIC vs TCP for Video Streaming QoE Provisioning. In *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 7–12.
- [27] Michael Seufert, Nikolas Wehner, and Pedro Casas. 2018. Studying the Impact of HAS QoE Factors on the Standardized QoE Model P.1203. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1636–1641.
- [28] Michael Seufert, Nikolas Wehner, and Pedro Casas. 2019. A Fair Share for All: TCP-inspired Adaptation Logic for QoE Fairness Among Heterogeneous HTTP Adaptive Video Streaming Clients. *IEEE Transactions on Network and Service Management* 16, 2 (2019), 475–488.
- [29] Michael Seufert, Nikolas Wehner, Florian Wamser, Pedro Casas, Alessandro D'Alconzo, and Phuoc Tran-Gia. 2017. Unsupervised QoE Field Study for Mobile YouTube Video Streaming with YoMoApp. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 1–6.
- [30] Michael Seufert, Bernd Zeidler, Florian Wamser, Theodoros Karagioules, Dimitrios Tsilimantou, Frank Loh, Phuoc Tran-Gia, and Stefan Valentin. 2018. A Wrapper for Automatic Measurements with YouTube's Native Android App. In *2018 Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–8.
- [31] Seufert, Michael and Wehner, Nikolas and Wieser, Viktoria and Casas, Pedro and Capdehourat, Germán. 2020. Mind the (QoE) Gap: On the Incompatibility of Web and Video QoE Models in the Wild. In *2020 16th International Conference on Network and Service Management (CNSM)*. IEEE, 1–5.
- [32] Bruce Spang, Shravya Kunamalla, Renata Teixeira, Te-Yuan Huang, Grenville Armitage, Ramesh Johari, and Nick McKeown. 2023. Sammy: Smoothing Video Traffic to Be A Friendly Internet Neighbor. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 754–768.
- [33] Barbara Staehle, Matthias Hirth, Rastin Pries, Florian Wamser, and Dirk Staehle. 2010. YoMo: A YouTube Application Comfort Monitoring Tool. *New Dimensions in the Assessment and Support of Quality of Experience for Multimedia Applications, Tampere, Finland* 6 (2010).
- [34] Florian Wamser, Andreas Blenk, Michael Seufert, Thomas Zimmer, Wolfgang Kellerer, and Phuoc Tran-Gia. 2015. Modelling and Performance Analysis of Application-aware Resource Management. *International Journal of Network Management* 25, 4 (2015), 223–241.
- [35] Florian Wamser, Michael Seufert, Pedro Casas, Ralf Irmer, Phuoc Tran-Gia, and Raimund Schatz. 2015. YoMoApp: A Tool for Analyzing QoE of YouTube HTTP Adaptive Streaming in Mobile Networks. In *2015 European Conference on Networks and Communications (EuCNC)*, pages=239–243. IEEE.
- [36] Shichang Xu, Eric Petajan, Subhabrata Sen, and Z Morley Mao. 2020. What You See Is What You Get: Measure ABR Video Streaming QoE via On-device Screen Recording. In *Proceedings of the 30th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*. 60–66.
- [37] Yali Yuan, Weijun Wang, Yuhan Wang, Sripriya S Adhatarao, Bangbang Ren, Kai Zheng, and Xiaoming Fu. 2022. VSiM: Improving QoE Fairness for Video Streaming in Mobile Environments. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 1309–1318.
- [38] Guanghui Zhang, Ke Liu, Haibo Hu, and Jing Guo. 2021. Short Video Streaming with Data Wastage Awareness. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.
- [39] Haodan Zhang, Yixuan Ban, Xingcong Zhang, Zongming Guo, Zhimin Xu, Shengbin Meng, Junlin Li, and Yue Wang. 2020. APL: Adaptive Preloading of Short Video with Lyapunov Optimization. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 13–16.
- [40] Yuming Zhang, Yan Liu, Lingfeng Guo, and Jack YB Lee. 2022. Measurement of A Large-scale Short-video Service Over Mobile and Wireless Networks. *IEEE Transactions on Mobile Computing* 22, 6 (2022), 3472–3488.
- [41] Chao Zhou, Yixuan Ban, Yangchao Zhao, Liang Guo, and Bing Yu. 2022. PDAS: Probability-driven Adaptive Streaming for Short Video. In *Proceedings of the 30th ACM International Conference on Multimedia*. 7021–7025.
- [42] Shangyue Zhu, Theo Karagioules, Emir Halepovic, Alamin Mohammed, and Aaron D Striegel. 2022. Swipe Along: A Measurement Study of Short Video Services. In *Proceedings of the 13th ACM Multimedia Systems Conference*. 123–135.