PDAS: Probability-Driven Adaptive Streaming for Short Video

Chao Zhou
Kuaishou Technology
Beijing, China
zhouchao@kuaishou.com

Yixuan Ban
Kuaishou Technology
Beijing, China
banyixuan@kuaishou.com

Yangchao Zhao
Kuaishou Technology
Beijing, China
zhaoyangchao03@kuaishou.com

Liang Guo
Kuaishou Technology
Beijing, China
guoliang@kuaishou.com

Bing Yu
Kuaishou Technology
Beijing, China
yubing@kuaishou.com

ABSTRACT

To improve Quality of Experience (QoE) for short video applications, most commercial companies adopt preloading and adaptive streaming technologies concurrently. Though preloading can reduce rebuffering, it may greatly waste bandwidth if the downloaded video chunks are not played. Also, each short video's downloading competes against others, which makes the existing adaptive streaming technologies fail to optimize the QoE for all videos. In this paper, we propose PDAS, a Probability-Driven Adaptive Streaming framework, to minimize the bandwidth waste while guaranteeing QoE simultaneously. We formulate PDAS into an optimization problem, where a probabilistic model is designed to describe the swiping events. Then, the maximum preload size is controlled by the proposed probability-driven max-buffer model, which reduces the bandwidth waste by proactively *sleeping*. At last, the optimization problem is solved by jointly deciding the preload order and preload bitrate. Extensive experimental results demonstrate that PDAS achieves almost 22.34% gains on QoE and 22.80% reductions on bandwidth usage against the existing methods. As for online evaluation, PDAS ranks *first* in the *ACM MM 2022 Grand Challenge: Short Video Streaming*.

CCS CONCEPTS

• **Information systems → Multimedia streaming**.

KEYWORDS

Short video streaming, probability-driven QoE formulation

ACM Reference Format:
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 (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3503161.3551571

1 INTRODUCTION

In short video applications like Kuaishou [3] and YouTube Shorts, users watch videos in the mode of swiping and watching. To ensure the QoE, the current video and videos in the recommendation queue are generally preloaded. Besides, adaptive bitrate (ABR) streaming technology is also widely adopted to adapt to diverse network conditions. Short video companies have paid hundreds of millions of dollars on bandwidth each year, while 40% of that has been wasted due to the user's swiping operation, as reported in [14, 15].

However, how to save bandwidth overhead without sacrificing the QoE still faces great challenges. First, *how to determine the preload size*? The preloading technology that can reduce the rebuffering time [10, 11] has been widely adopted by the short video platforms [12]. However, it may lead to huge bandwidth waste if the preloaded contents are not watched. Second, *how to schedule the preload order*? The current video and videos in the recommendation queue compete with each other for the limited network resources, which means the preloading operation always reduces some videos' rebuffering at the expense of increasing others. The swiping and watching mode makes it more challenging since the preloaded videos may be swiped away and thus resulting in more bandwidth waste. At last, *how to select the preload bitrate*? In addition to the challenges in typical ABR algorithms (e.g., bandwidth estimation, buffer prediction, bitrate selection's cascading effects, etc.), the viewing probability of each chunk should be taken into consideration to accurately evaluate each chunk's expected QoE and bandwidth consumption.

Unfortunately, off-the-shelf short video streaming algorithms fall short of resolving the tasks. Our prior work SR2A [17] selects each video's bitrate without considering bandwidth waste, while DUASVS [15] empirically sets the max buffer size to save bandwidth only, without any QoE performance guarantee. Although APL [16] builds the QoE and waste model together to instruct the downloads, it simply assumes that the user's swiping probability satisfies a *fixed* distribution over all videos, while QASVS [5] and LiveClip [6] regard the swiping events as *random* behaviors. However, substantial differences in the swipe distributions across different videos have been observed [8]. The missing of the user's swiping probability model hinders the QoE and bandwidth saving improvement.

In this work, we propose PDAS, a Probability-Driven Adaptive Streaming framework for short video platforms, where the effect of preload *size*, *order*, and *bitrate* on bandwidth waste and QoE is systematically considered. Specifically, we formulate PDAS into an optimization problem, where a probabilistic model is designed

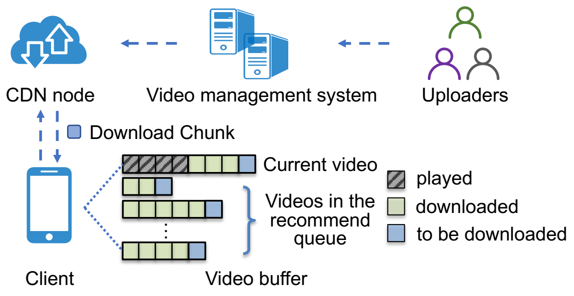

Figure 1: Short Video Streaming System Overview

to describe the swiping events by analyzing the user's retention ratio from Kuaishou [3], one of the largest short video communities worldwide (§3.2). Then, a probability-driven max-buffer model is proposed to control the maximum preload *size* of each video, which explicitly saves bandwidth consumption by proactively stopping requesting new contents (i.e., *sleep*) when the current buffer length exceeds the target limit (§3.3). Based on that, we solve the optimization problem for the non-sleeping videos to decide the preload *order* and *bitrate* on each download step (§3.4). We implement PDAS and compare it with the state-of-the-art streaming algorithms on trace-driven emulations. The results demonstrate that PDAS achieves almost 22.34% gains on QoE and 22.80% reductions on bandwidth usage against the existing methods. Also, PDAS ranks first on each online evaluation set provided by the ACM MM 2022 Grand Challenge: Short Video Streaming [2].

2 SYSTEM OVERVIEW

As shown in Fig. 1, in short video applications, the videos are uploaded to the video management system and Content Delivery Network (CDN) node to serve the audience. On the client side, in addition to the currently playing video, there are several videos waiting in the recommendation queue. On each step, the client can pick one chunk from the videos to download. To optimize the bandwidth utilization and QoE concurrently, PDAS firstly formulates the swiping probability of each chunk according to the offline collected viewing retention ratio and real-time playback progress (§3.2). Then, the probability is leveraged to determine the max-buffer size (§3.3). For videos without exceeding the buffer limit, their chunks encoded at different bitrates are fed into the QoE-waste aware optimization problem, and the chunk achieving the highest optimization score is picked to be downloaded (§3.4). Or else, the client will sleep for a fixed duration to save bandwidth.

3 DESIGN OF PDAS

3.1 Problem Formulation

As discussed in [9, 13], the user's QoE is mainly determined by the average video quality Φ, average quality variation Ψ, and the rebuffering duration Γ. To support ABR streaming [4], each video i is encoded into N bitrate levels and then cropped into chunks with T seconds. We denote $r_{i,m}$ as the encoding bitrate of the m-th chunk (it particularly denotes the to be actually downloaded chunk in this paper) in video i, where $r_{i,m} \in \mathcal{R}^N$. Then, the QoE of video

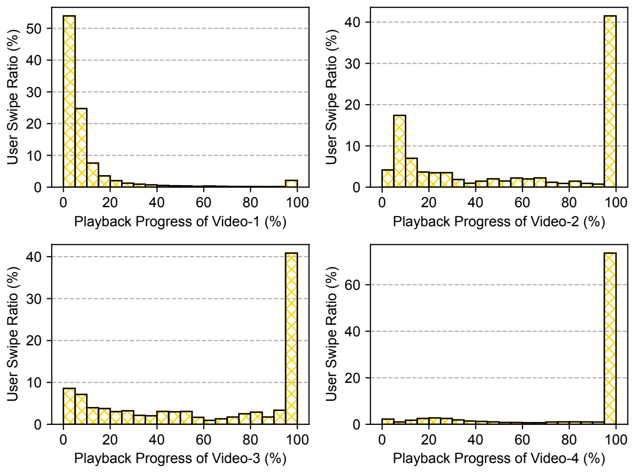

Figure 2: User Swipe Ratio Distribution

i can be formulated as

$$QoE_i = \sum_m (w_1 \cdot \Phi(r_{i,m}) - w_2 \cdot \Psi(r_{i,m}) - w_3 \cdot \Gamma(r_{i,m})). \quad (1)$$

The bandwidth consumed by downloading video i is

$$Cost_i = \sum_m S(r_{i,m}), \quad (2)$$

where $S(r_{i,m})$ is the actual encoding size of bitrate $r_{i,m}$. Then, to minimize the bandwidth waste while guaranteeing QoE concurrently, the object of PDAS can be formulated as

$$\underset{\{r_{i,m} \in \mathcal{R}^N\}}{\arg\max} \sum_i (QoE_i - w_4 \cdot Cost_i), \quad (3)$$

where w_1, w_2, w_3, and w_4 are respectively set to 1, 1, 1.85, and 0.5 as the Grand Challenge required.

3.2 Probabilistic Model of User Retention

The expected QoE and bandwidth waste of each chunk are strongly related to the user's swiping behavior. However, the behavior is usually considered to be completely random or fixed in the existing streaming system [6, 15, 16], although it is not actually. To verify this, we randomly select 4 highly viewed videos including game, adventure, pet, and makeup from Kuaishou [3]. Then we plot the swipe ratio of users on different playback progress (5% per bar) in Fig. 2. As shown, there are significant differences in the user swiping ratio distributions. For example, about 50% of the users will swipe it away within the first 5% of the time in video-1, whereas almost 80% of the users in video-4 will finish their playback.

To mathematically represent the user's swiping behavior and thus make better download decisions, we calculate the proportion of users who watch video i from the first chunk to the m-th chunk as the average retention rate $H_{i,m}$ of chunk m on video i, where $0 < H_{i,m} \leq 1$. Then, assuming the current playback progress is m_c (denoted by the chunk index), the probability of watching the i-th video from m_c until m is

$$p_{i,m}(m_c) = \begin{cases} \frac{H_{i,m}}{H_{i,m_c}}, & m > m_c \\ 1, & m \leq m_c \end{cases}. \quad (4)$$

Note that each video's average retention rate $H_{i,m}$ is provided in the Grand Challenge online evaluation platform, and it can be easily collected on the real short video streaming applications.

3.3 Max Buffer Model

To reduce the bandwidth waste, an effective way is to control the preload size, i.e., stop downloading when the buffered video exceeds the max buffer threshold. However, a small threshold may lead to far more rebuffering events in poor network conditions.

To solve this, we present a probability-driven max-buffer model to control the threshold in PDAS. Specifically, for chunk m ($m > m_c$) to be downloaded in video i, we define the dynamic max buffer threshold as

$$b_{i,m}^{max} = \max\{p_{i,m}(m_c) \cdot T_{i,m}^{max}, b_i^{th}\}, \tag{5}$$

where $T_{i,m}^{max}$ is the time consumed to download the m-th chunk with the maximal bitrate of video i. Considering the user may swipe the video away at any time, we shrink the max-buffer by the probability of watching the video from the current playback progress m_c to chunk m, i.e., $p_{i,m}(m_c)$, to save bandwidth,

Moreover, we set the lower bound of $b_{i,m}^{max}$ as b_i^{th}, which is a buffer threshold to avoid rebuffering when bandwidth suddenly drops. As we know, rebuffering is affected by both bandwidth and the time that the video is played (playback order). Specifically, when bandwidth increases, the risk of rebuffering decreases. Meanwhile, the farther the video is from the currently playing video, the less rebuffering it will encounter. Thus, for the i-th ($i \geq i_c$) video, the b_i^{th} can be given by an exponential function as

$$b_i^{th} = \epsilon \cdot e^{-\lambda_1 C - \lambda_2 (i - i_c)}, \tag{6}$$

where $i - i_c$ is the distance between video i and video i_c, C is the estimated bandwidth, ϵ, λ_1 and λ_2 are set as 3.5, 0.3, and 0.15.

Based on the formulations above, the dynamical max-buffer $b_{i,m}^{max}$ can be determined, which is affected by the retention probability, playback order, and bandwidth together. Especially, if the buffered duration exceeds the max buffer for all videos, the client would sleep for τ seconds to save bandwidth, where $\tau = 0.05$ typically.

3.4 QoE-Waste Aware Bitrate Adapter

To jointly determine the preload order and bitrate precisely, we give the details of the objective in (3). Specifically, the expected quality $\Phi(r_{i,m})$ of downloading chunk m with $r_{i,m}$ is determined by the quality of $r_{i,m}$ and the retention possibility $p_{i,m}(m_c)$:

$$\Phi(r_{i,m}) = p_{i,m}(m_c) \cdot q(r_{i,m}), \tag{7}$$

where $q(r_{i,m})$ is the quality metric, which can be calculated via Peak Signal-to-Noise Ratio (PSNR), Video Multimethod Assessment Fusion (VMAF) [7], and bitrate level [13]. Accordingly, the average quality variation $\Psi(r_{i,m})$ is given as

$$\Psi(r_{i,m}) = p_{i,m}(m_c) \cdot |q(r_{i,m}) - q(r_{i,m-1})|. \tag{8}$$

For video i, we denote the current buffer occupancy is b_i, the time of downloading the m-th chunk with bitrate $r_{i,m}$ is $T(r_{i,m}) = \frac{S(r_{i,m})}{C}$, and k chunks would be played with

$$k = \lceil \frac{T(r_{i,m})}{T} \rceil \tag{9}$$

Algorithm 1 PDAS Algorithm Workflow

Input: Last download size $S(r'_{i,m})$; last download time $T(r'_{i,m})$; buffer $\{b_i\}$; playback progress m_c; user retention rate $\{H_{i,m}\}$.
Output: The best chunk $r^*_{i,m}$ to download or the sleep time τ.

```
1:  Update estimated bandwidth $C$ by $S(r'_{i,m})$ and $T(r'_{i,m})$ as [13].
2:  Update retention probability $\{p_{i,m}(m_c)\}$ by (4).
3:  Update max-buffer $\{b^{max}_{i,m}\}$ by (5)-(6).
4:  Initialize $r^*_{i,m} \leftarrow 0$ and score of (3) as $U^* \leftarrow -\infty$.
5:  for video $i \geq i_c$, $b_i \leq b^{max}_{i,m}$ do
6:      Run $K$-step RobustMPC in [13] with (7)-(13)
7:      Record the best bitrate $r_{i,m}$ and the corresponding $U_i$ in (3).
8:      if $U_i \geq U^*$ then
9:          $U^* \leftarrow U_i$, $r^*_{i,m} \leftarrow r_{i,m}$.
10:     end if
11: end for
12: Return $r^*_{i,m}$ if $U_i \neq -\infty$ else Return sleep time of $\tau$ seconds.
```

Then, assuming the user swipes from video i_c to video j ($i_c \leq j$) during the download of $r_{i,m}$, the expected rebuffering duration on video j can be represented by

$$\Gamma_j(r_{i,m}) = p_{j,z+k}(z) \cdot \max\{T(r_{i,m}) - b_j, 0\}, \tag{10}$$

where

$$z = \begin{cases} m_c, & j = i_c \\ 0, & j > i_c \end{cases}, \tag{11}$$

$p_{j,z+k}(z)$ denotes the probability that the user has not swiped away from video j until the m-th chunk of video i with bitrate $r_{i,m}$ is completely downloaded. Especially, we calculate each video's rebuffering time by assuming video j is the only playing one during the whole download process, i.e., the expected playing chunk number k for each player is constant. This assumption is not precise enough since the user can play any of the videos between i_c and j and thus could reduce the playback and rebuffering time on video j. Nevertheless, the approximation is robust for the real network conditions [13] because the calculated rebuffering time is an upper bound actually. Meanwhile, we calculate the estimated bandwidth by RobustMPC [13], which normalizes the bandwidth by the max prediction error of the sliding window to further improve the robustness on fluctuating network conditions.

To represent the total rebuffering of downloading $r_{i,m}$, we approximately calculate the probability $P_{i_c,i}$ that the user swipes from current playing video i_c to video j ($i_c \leq j$) during the download as

$$P_{i_c,j} = \begin{cases} \prod_{l=i_c}^{j-1} (1 - p_{l,z+k}(z)), & j > i_c \\ 1, & j = i_c \end{cases}. \tag{12}$$

Then the total rebuffering expectation is summarized as

$$\Gamma(r_{i,m}) = \sum_{j=i_c}^{L} P_{i_c,j} \cdot \Gamma_j(r_{i,m}). \tag{13}$$

With the above formulations, Algo. 1 shows the solution. At each download step, we calculate the bandwidth C, retention probability $\{p_{i,m}(m_c)\}$ and the max buffer $\{b^{max}_{i,m}\}$ first. Then, we loop the videos with buffer $b_i \leq b^{max}_{i,m}$ to conduct K-step RobustMPC as

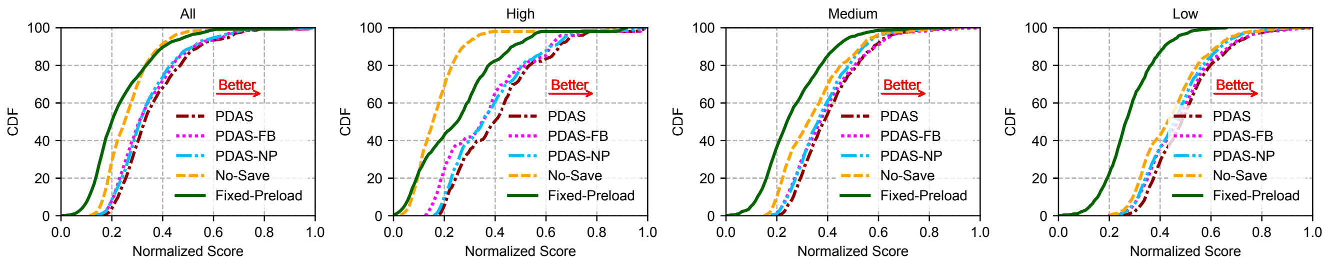

Figure 3: Performance on total score

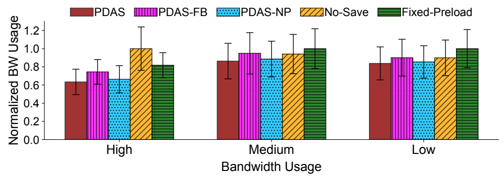

Figure 4: Performance on bandwidth usage

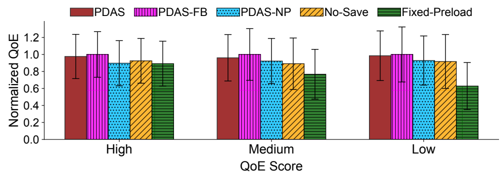

Figure 5: Performance on QoE

[13] to find the best $r_{i,m}$ and U_i. Finally, if all the video buffers are larger than the $b_{i,m}^{max}$, we return the sleep time of τ seconds to save bandwidth, or else we return the $r_{i,m}$ achieving the U^*.

4 EVALUATION

4.1 Methodology

Simulation Testbed. We use the trace-driven simulator provided by the Grand Challenge [2] to evaluate PDAS, which simulates the user's playing and swiping behavior by sampling the offline video retention rate table and the bandwidth traces. In this simulator, there are 4 players in the recommendation queue, i.e., 5 players in total. Each video is encoded into $N = 3$ representations on different bitrates and further cropped into chunks with $T = 1$ second. The test cases are divided into 3 scenarios: 1) High bandwidth, 2) Medium bandwidth, and 3) Low bandwidth. For each case, we sample 50 playback traces, multiplied by 20 network traces as the Grand Challenge required, to evaluate the performance.

Compared Methods. We compare PDAS with two baselines and two PDAS ablation methods. **No-Save** and **Fixed-Preload** both preload videos after the current video is downloaded completely, which are the default baseline of the Grand Challenge [1]. No-Save will keep preloading each video with 800KBytes per step [16] and Fixed-Preload only preloads with 4 chunks at most. **PDAS-FB** is PDAS with a fixed max-buffer like Fixed-Preload. **PDAS-NP** is PDAS without retention probability modeling.

4.2 Offline Evaluation

Fig. 3 shows the CDF of the average utility score across all scenarios respectively. PDAS outperforms existing baselines over all considered video scenarios, with the improvements on QoE (Fig. 5) and waste (Fig. 4) by 6.62% and 22.80% on No-Save, 22.34% and 18.30% on Fixed-Preload. It makes sense since PDAS leverages a probability-driven max-buffer controller to avoid downloading redundant data and utilizes the QoE-waste aware bitrate adaption to improve the user's QoE simultaneously. Moreover, we find that PDAS and PDAS-NP achieve minimal bandwidth usage in all test cases by leveraging the buffer controller in §3.3. No-Save consumes the highest bandwidth especially in the High bandwidth scenario since it never sleeps. Especially, the Fixed-Preload preloads chunks with the highest bitrate when the buffer exceeds 2 seconds [2], thus its bandwidth waste is severe in medium and low bandwidth, also the unwatched contents hinder the playback video to perceive high QoE. Further, we notice that PDAS and PDAS-FB achieve the maximum QoE because they both adopt the probability-driven bitrate adaption in §3.4. PDAS-FB even performs slightly better than PDAS as preloading more data can reduce rebuffering.

Table 1: Average Performance Score in Online Competition.

Team Name	Round-1	Round-2	Round-3
Kwai 2022	**2827.66 (1st)**	**2491.13 (1st)**	**3287.05 (1st)**
No1	2637.20 (2nd)	2255.40 (2nd)	2931.48 (2nd)
sky_light	2491.83 (5th)	2221.52 (3rd)	2843.02 (3rd)
EVA00	2490.37 (6th)	2208.21 (4th)	2833.55 (5th)
MC2	2490.12 (7th)	2208.17 (5th)	2837.05 (4th)

4.3 Online Competition

Beyond the offline evaluation, PDAS is well behaved in the unseen network conditions and user behaviors in the online competition. As shown in Tab. 1, our team (Kwai 2022) ranks first among 30 teams worldwide over all three rounds of evaluation [2]. We achieve an overall score of 2868.61, 10% ahead of the second, which further demonstrates the effectiveness and robustness of PDAS.

5 CONCLUSION

In this paper, we propose a novel probability-driven adaptive streaming approach, namely PDAS, to strike the best trade-off between QoE and bandwidth waste in short video applications. To the best of our knowledge, PDAS is the first full-fledged scheme that utilizes the user retention probability to construct the max-buffer model and the QoE-waste optimization objective. It concurrently determines the preload size, order, and bitrate for the client, and perceives the highest score in not only the offline evaluation, but also the online competition of the ACM MM 2022 Grand Challenge.

REFERENCES

[1] 2022. ACM Multimedia 2022 Grand Challenge Official Github. https://github.com/AItransCompetition/Short-Video-Streaming-Challenge. [Online; accessed 8-June-2022].

[2] 2022. ACM Multimedia 2022 Grand Challenge: Short video streaming. https://www.aitrans.online/MMGC2022/. (2022). [Online; accessed 8-June-2022].

[3] 2022. Kuaishou. https://www.kuaishou.com/. [Online; accessed 8-June-2022].

[4] ISO/IEC JTC1/SC29/WG11 W13533. MPEG DASH: The Standard for Multi-media Streaming over the Internet..

[5] Jing Guo and Guanghui Zhang. 2021. A Video-Quality Driven Strategy in Short Video Streaming. In *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 221–228.

[6] 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.

[7] 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).

[8] Zhuqi Li, Yaxiong Xie, Ravi Netravali, and Kyle Jamieson. 2022. Dashlet: Taming Swipe Uncertainty for Robust Short Video Streaming. *arXiv preprint arXiv:2204.12954* (2022).

[9] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 197–210.

[10] Dezhi Ran, Huadun Hong, Yang Chen, Bonan Ma, Yuanxing Zhang, Pengyu Zhao, and Kaigui Bian. 2020. Preference-Aware Dynamic Bitrate Adaptation for Mobile Short-Form Video Feed Streaming. *IEEE Access* 8 (2020), 220083–220094.

[11] Dezhi Ran, Yuanxing Zhang, Wenhan Zhang, and Kaigui Bian. 2020. SSR: Joint Optimization of Recommendation and Adaptive Bitrate Streaming for Short-form Video Feed. In *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 418–426.

[12] Zhu Shangyue, Karagioules Theo, Halepovic Emir, et al. 2022. Swipe Along: A Measurement Study of Short Video Services. *ACM on Multimedia Systems Conference (MMSys'22)* (2022).

[13] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 325–338.

[14] 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.

[15] Guanghui Zhang, Jie Zhang, Ke Liu, Jing Guo, Jack Lee, Haibo Hu, and Vaneet Aggarwal. 2022. DUASVS: A Mobile Data Saving Strategy in Short-form Video Streaming. *IEEE Transactions on Services Computing* (2022).

[16] Haodan Zhang, Yixuan Ban, Xinggong 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.

[17] Chao Zhou, Shucheng Zhong, Yufeng Geng, and Bing Yu. 2018. A Statistical-based Rate Adaptation Approach for Short Video Service. In *2018 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 1–4.