The following publication G. Zhang, K. Liu, H. Hu, V. Aggarwal and J. Y. B. Lee, "Post-Streaming Wastage Analysis – A Data Wastage Aware Framework in Mobile Video Streaming," in IEEE Transactions on Mobile Computing, vol. 22, no. 1, pp. 389-401, Jan. 2023 is available at https://doi.org/10.1109/TMC.2021.3069764.

Post-Streaming Wastage Analysis – A Data Wastage Aware Framework in Mobile Video Streaming

Guanghui Zhang, Ke Liu, Haibo Hu, *Senior Member*, *IEEE*,
Vaneet Aggarwal, *Senior Member*, *IEEE*, and Jack Y. B. Lee, *Senior Member*, *IEEE*

Abstract—Mobile video streaming is now ubiquitous among mobile users. This work investigates a less studied and yet significant problem in mobile video streaming – data wastage, i.e., some downloaded video data may not be played back but discarded by video players due to early departure or video skip, thus the bandwidth consumed in transferring them is wasted. Our measurements show that data wastage is significant in practice, e.g., 25.2%~51.7% of video data downloaded are in fact wasted. Moreover, substantial data wastage exists not only in current commercial streaming platforms, but also in state-of-the-art adaptive streaming systems proposed in the literature. This work develops a new Post-Streaming Wastage Analysis (PSWA) framework to tackle this problem by converting existing adaptive streaming algorithms into data wastage aware versions. PSWA enables the streaming vendors to explicitly control the tradeoff between data wastage and quality-of-experience (QoE). Extensive evaluations show that PSWA can reduce data wastage significantly (e.g., 80%) without any adverse impact on QoE. Moreover, it has strong robustness to perform consistently across a wide range of networks. PSWA can be readily implemented into current streaming platforms, and thus offers a practical solution to data wastage for mobile streaming services.

Index Terms—Video Streaming; Mobile Network; Data Wastage; Quality-of-experience.

1 Introduction

MOBILE video streaming has quickly become a key application in the mobile Internet [1]. For many mobile users, watching videos using their smartphone has become a daily activity. With so many sources of videos, it is not surprising that not all the videos are watched from start to finish. In fact, due to common viewing behaviors such as *early departure* and *video skip* (i.e., changing to a different playback point), a significant portion of videos were not watched completely by viewers [2-4]. For example, Finamore *et al.* [2] measured the video access logs on YouTube and found that 60% of videos were watched for no more than 20% of their whole duration. However, a side-effect of early departure and video skip is that some of the downloaded video data are *discarded* and the bandwidth consumed in transferring them is thus wasted. We call this *data wastage* in the rest of the paper.

At first glance, such data wastage may not appear to be a significant issue. However, current on-demand video streaming (VoD) has practically all migrated to some forms of HTTP-based bitrate adaptive transfer protocol (e.g., DASH [5]). Common to these protocols is the use of HTTP over TCP to transfer the video data as fast as TCP allows. Therefore, if the TCP throughput is higher than the selected video bitrate then the client will fetch video data ahead of their playback schedules and store them in the local buffer. This can improve streaming performance significantly, as the buffered data can be used to absorb mobile networks' bandwidth fluctuations to prevent playback rebuffering. However, the same fetch-ahead buffering mechanism would also increase data wastage significantly if the viewer terminates or skips video playback before all downloaded data are rendered.

Our measurements of existing adaptive streaming algorithms showed that 25.2%~51.7% of video data downloaded were wasted. This level of data wastage has two far-reaching consequences. First, today's mobile data services purchased by users generally have a hard data cap, e.g., 10 GB per month [6]. If the data usage exceeds the given data quota, mobile users have to purchase additional data quota at a much higher price. Therefore, given the significant data wastage, a substantial portion of the data quota would be wasted in transferring video data which are never watched. Second, data wastage consumes precious bandwidth resources from the streaming vendor's network (e.g., CDN), which are often charged by volume of data transferred. Given the immense cost of the infrastructure, even a tiny percentage of wasted bandwidth can be financially significant to streaming vendors. For exam-

- G. Zhang and J. Y. B. Lee are with the Department of Information Engineering, the Chinese University of Hong Kong, Shatin, NT, Hong Kong SAR. E-mail: {ghzhang, yblee}@ie.cuhk.edu.hk
- K. Liu is with the State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences (UCAS), Beijing, China. E-mail: liuke@ict.ac.cn
- H. Hu is with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Kowloon, Hong Kong and PolyU Shenzhen Research Institute. E-mail: haibo.hu@polyu.edu.hk
- V. Aggarwal is with the School of Industrial Engineering, Purdue University, West Lafayette, IN 47907 USA and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. E-mail: vaneet@purdue.edu

ple, Chen *et al.* [7] measured that the cost due to data wastage could be tens to hundreds of millions of dollars each year.

One method to reduce data wastage is to limit the video client buffer size. Taking it to the extreme, if the player buffers no more than one video segment at any time then the worst-case data wastage will only be one segment. However, the client buffer exists for an important reason – to buffer data such that video playback can be sustained during periods of low bandwidth so that playback rebuffering can be avoided. Too small a buffer will likely lead to frequent rebuffering and significant Quality-of-Experience (QoE) degradation, which can be an even bigger problem than data wastage. This is especially important in the mobile network where rapid and substantial bandwidth fluctuations are the norm rather than the exception.

Therefore, the fundamental question is whether a feasible tradeoff between QoE and data wastage exists in today's mobile networks, and if so, how to achieve a desired wastage-QoE tradeoff in a streaming platform. This work is the first attempt to provide an answer to these questions by developing a new Post-Streaming Wastage Analysis (PSWA) framework to allow the streaming vendor to explicitly control the tradeoff between data wastage and QoE. Specifically, PSWA introduces two wastage-aware parameters that can be easily incorporated into existing adaptive streaming algorithms, so that fine-grained control of wastage-QoE tradeoffs can be enabled. By analyzing the streaming trace data from past video sessions, PSWA automatically optimizes the wastage-aware parameters and then applies them to future video sessions to minimize data wastage while maintaining high QoE.

Extensive evaluations showed that PSWA can reduce data wastage by 31.6%~79.9% even without any QoE loss. In addition, it could reduce data wastage even further by small tradeoffs in QoE (e.g., 4% drop in QoE improves data wastage reduction to 44.4%~90.2%). Moreover, PSWA performs consistently across a wide range of networks. Therefore, it offers an immediate and practical solution to reduce data wastage in current and future streaming platforms.

This work has three major contributions. First, since data wastage and QoE are inherently conflicting objectives, reducing wastage may result in QoE loss. However, QoE is critical to streaming services and the tolerance for QoE loss differs among different streaming vendors. PSWA addresses this challenge by providing streaming vendors with an interface called acceptable QoE loss ratio to allow them to specify their QoE preference. Specifically, they can set the QoE loss ratio to any values within 0%~100% where 0% means no QoE loss. According to the ratio, PSWA minimizes data wastage and meanwhile ensures the actual QoE degradation not exceed the ratio. To the best of our knowledge, PSWA is the first system that can control data wastage based on the streaming vendor's QoE preference.

Second, PSWA breaks the one-size-fits-all approach commonly adopted by the existing data wastage solutions [7-11] and optimizes wastage-aware parameters according to the specific network condition. This enables PSWA to not only outperform the existing approaches significantly, but also have strong robustness to achieve consistent performance across a wide range of network environments.

Last but not least, PSWA is designed to complement (as opposed to replacing) the existing adaptive streaming algorithms by converting them into wastage-aware versions while keeping their original adaptation logic intact. This offers an immediate and ready solution for the streaming platforms already in service. Although this work focuses on adaptive on-demand streaming, PSWA is a generic framework that can potentially be extended to other streaming services, such as non-adaptive streaming, 360-degree video streaming, live streaming, etc.

The rest of the paper is organized as follows: Section 2 reviews the related work; Section 3 investigates the data wastage problem in mobile video streaming; Section 4 presents the design of the PSWA framework; Section 5 evaluates the performance of PSWA by trace-driven simulations and real experiments, and Section 6 summarizes the study and outlines some future work.

2 Related Work

Much work has been done in video streaming in recent years. A comprehensive review of the area is beyond the scope of this work. We refer interested readers to the studies by Seufert *et al*. [12], Juluri *et al*. [13], Kua *et al*. [14] and Bentaleb *et al*. [15] for survey and comparison of existing streaming algorithms.

Existing streaming algorithms were primarily designed to improve QoE performance. Much of the intelligence of a streaming algorithm is in selecting the most appropriate video bitrate level from the ones available at the server such that playback continuity and high video quality can be maintained. As data wastage does *not* impact QoE directly, it is no surprise that data wastage is often neglected in the design of streaming algorithms. Nevertheless, with the almost ubiquitous deployment of HTTP-based video streaming, data wastage can no longer be an afterthought. An early measurement study by Finamore *et al*. [2] analyzed YouTube and found that data wastage is significant, e.g., during peak hours, 25%~39% of bandwidth was wasted by desktop users and 35%~48% by mobile users.

Chen *et al*. [7] looked into the data wastage problem in Tencent Video [16] and found that over 20% of bandwidth was wasted due to video data delivered but unwatched. To reduce data wastage, they developed a server-side Behavior-Based (henceforth called BB) streaming strategy. BB was designed for the scenario where the network is already fully utilized. It reduced data wastage through limiting the transmission rate to 1.05 times of the video bitrate (as opposed to as fast as TCP allows) during the viewing browsing phase (this phase generally exists at the beginning of videos with high departure rate [17]). The bandwidth saved in this phase can then be reallocated to other streaming clients to improve their QoE. However, BB was designed only for non-adaptive streaming so it may not be directly applicable to today's adaptive streaming platforms (e.g., DASH [5]).

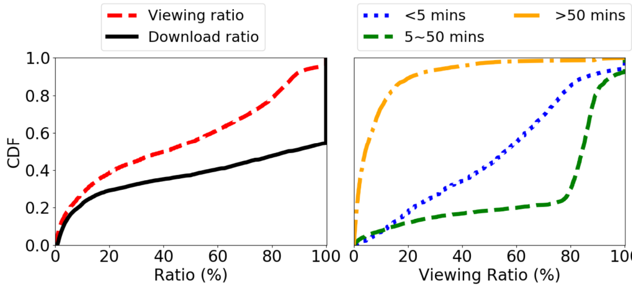


Fig. 1. Statistics for viewing ratio and download ratio.

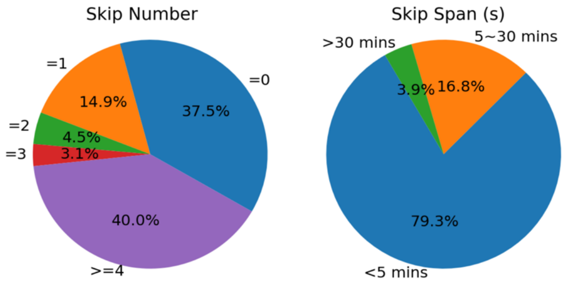


Fig. 2. Statistics for video skip number and skip span.

In a recent study, Yarnagula *et al.* [8] proposed SARA to reduce data wastage for adaptive video streaming. SARA was deployed in the video clients and designed for reducing data wastage through limiting the amount of data in the buffer with a pre-defined buffer threshold (i.e., 20s). Specifically, when the client buffer occupancy reaches the buffer threshold, the request for downloading the next segment will be delayed until the buffer occupancy falls below the threshold. In another study, Chen *et al.* [9] proposed an energy-aware rate adaptation algorithm that controls data wastage in the same way as SARA but sets the buffer threshold to 30s. However, our empirical study (c.f. Section 3.2) showed that merely limiting the buffer size would lead to more rebuffering events which degrade QoE performance.

In another direction, both Li *et al.* [10] and Huang *et al.* [11] proposed to use Lyapunov optimization theory to design bandwidth allocation strategies for the base station with the goal to reduce the total data wastage for all mobile users served by the base station. However, in practice, their proposed strategies require mobile operators to modify the link-layer implementation of the base stations, which is far from simple in today's mobile infrastructures.

In comparison, the PSWA proposed in this study tackles all the limitations in these existing solutions and is able to effectively control data wastage in mobile networks.

3 Data Wastage in Mobile Video Streaming

In this section, we measure data wastage in HTTP-based on-demand streaming (VoD). We first investigate the two common viewing behaviors (early departure and video skip) and then employ trace-driven simulation to measure data wastage from state-of-the-art adaptive streaming algorithms.

3.1 Early Departure and Video Skip

We first look into *early departure* through a real-world empirical trace dataset [18]. The notion of data wastage is that some downloaded video data are not watched but discarded. Therefore, to measure data wastage, we should first measure the proportion of each video being watched and downloaded at the time of early departure. In the dataset, for video session i, $0 \le i < N$, we obtained the video physical duration, denoted by L_i, the amount/duration of video data downloaded, denoted by D_i, and the viewing duration, denoted by V_i.

To quantify early departure, we define *viewing ratio* ϕ_i as the ratio of video played back (in duration) to the video physical duration for video session i, i.e.,

$$\phi_i = V_i / L_i \tag{1}$$

Similarly, we define download ratio θ_i as the ratio of video downloaded (in duration) to the video physical duration, i.e.,

$$\theta_i = D_i / L_i \tag{2}$$

The left chart in Fig. 1 plots the distributions of the two ratios in the empirical dataset. It is evident that a significant proportion of video sessions ended early, with an overall average viewing ratio of 42.6%. In comparison, the download ratio is substantially higher, with an overall average of 63.1%. This suggests that a significant proportion of the video data was downloaded but not played back. We further divided all the video sessions into three subsets based on their video physical duration, i.e., short (<5 mins), medium (5~50 mins), and long (>50 mins), and then plotted their viewing ratio distribution in the right chart in Fig. 1.

We observed that their viewing ratios differ significantly. For example, viewers tend to leave relatively early when watching long-length videos (i.e., >50 minutes), whereas tend to watch completely when watching medium-length videos (i.e., 5~50 minutes). More detailed analysis for early departure can be found in Appendix A.1.

Next, we investigate *video skip* using an empirical model from [17]. The left chart in Fig. 2 plots the proportion of the mean skip number in each video session. We can observe that 62.5% of video sessions have video skips (i.e., except "= 0") and the proportion of skip number ">=4" is significantly higher than others. This is intuitive because if a viewer is not interested in the current video content, the viewer will naturally skip several times to keep looking for points of interest. The right graph in Fig.2 shows the proportion of skip span. The key observation is that nearly 80% of the skips are within 5 mins, and the proportion of long skip span (>30 min) is very small. Overall, in addition to early departure, video skip is also a very common viewer behavior that can cause data wastage. Next we will measure data wastage in streaming platforms based on these viewer behavior models.

Table 1
Statistics of Seven Throughput Trace Datasets.

Characteristics	Dataset #1	#2	#3	#4	#5	#6	#7
Throughput (Mbps)	5.57	4.71	3.29	2.87	1.21	12.1	3.12
Coefficient of Variation	0.44	0.39	0.74	0.53	0.83	0.69	0.59
Network type	3G	3G	3G	3G	3G	LTE	WiFi
Collection location	L1	L1	L2	L3	L4	L5	L6
Service provider	S1	S2	S1	S1	S3	S2	S4

3.2 Data Wastage Measurement

We employed trace-driven simulations to measure data wastage in realistic network settings where the simulator replicates the bottleneck link by replaying TCP throughput trace data obtained from real production mobile networks. We used a total of 60 weeks of TCP throughput trace data (~ 100,000 video sessions) covering 3G, 4G/LTE and Wi-Fi networks. The trace data are publicly available [20-22] and we summarized their key statistics in Table 1. Viewing behavior traces (e.g., early departure, video skip) were derived from the empirical datasets introduced in Section 3.1. The available video bitrates follow the Apple profile [19] augmented by four additional bitrates at 10 Mbps, 12 Mbps, 16 Mbps, and 20 Mbps. The rest of the streaming parameters are summarized in Table 2. Please refer to Appendix A.2 for more details of the simulation settings.

We implemented seven state-of-the-art streaming algorithms which include two throughput-based bitrate adaptive algorithms – LBG [23] and Stagefright [24], two buffer-based bitrate adaptive algorithms – BBA [25] and SARA [8], two hybrid throughput-buffer-based bitrate adaptive algorithms – RobustMPC (henceforth called MPC) [26] and Pensieve [27], and one non-adaptive algorithm BB [7]. It's worth noting that SARA and BB were originally designed with controlling data wastage in mind while all others were non-wastage-aware algorithms.

To quantify data wastage, we define a metric to compute the amount of data wastage in video session i, denoted by W_i, from the difference between video data downloaded and viewed:

$$W_i = \sum_{\forall d_{i,j}>0} d_{i,j} - \sum_{\forall v_{i,j}>0} s_{i,j} \frac{v_{i,j}}{l_{i,j}} \quad (3)$$

where $d_{i,j}$, $s_{i,j}$, $l_{i,j}$, $v_{i,j}$ are the downloaded data amount, segment size, full segment duration, segment duration viewed for segment j respectively. Similarly, we can compute the ratio of data wastage for session i, denoted by R_i, from

$$R_i = 1 - \sum_{\forall v_{i,j}>0} s_{i,j} \frac{v_{i,j}}{l_{i,j}} \Big/ \sum_{\forall d_{i,j}>0} d_{i,j} \quad (4)$$

Table 2
Evaluation Settings.

Parameters	Values
Bitrate profile	{0.2, 0.4, 0.8, 1.2, 2.2, 3.3, 5.0, 6.5, 8.6, 10, 12, 16, 20} Mbps [19]
Segment duration	2s
Video duration	Empirical distribution (30s to 10800s)
Session number	~ 100,000
Initial bitrate	0.2 Mbps

In addition to data wastage, for video session i, we also measured mean video bitrate – defined as the average bitrate selected, mean buffer occupancy – defined as the average buffer level, rebuffering duration – defined as the total time at which playback is suspended due to client buffer underflow, rebuffering frequency – defined as the total number of rebuffering events, and QoE – calculated by the QoE function proposed by Mao *et al.* [27]:

$$Q_i = \frac{1}{K}\left(\sum_{k=0}^{K-1} \vartheta_{i,k} - \sum_{k=1}^{K-1} \left|\vartheta_{i,k} - \vartheta_{i,k-1}\right| - 2.66 \times Z_i\right) \quad (5)$$

where Z_i is the rebuffering duration, K is the total number of segments in video session i and $\vartheta_{i,k}$ is the video quality calculated by

$$\vartheta_{i,k} = \log(r_{i,k}/r_{\min}) \quad (6)$$

where $r_{i,k}$ is the bitrate selected for segment k and r_{min} is the lowest available bitrate in the profile. Note that the coefficient of Z_i (i.e., 2.66) follows Mao *et al.* [27].

Table 3 summarizes the evaluation results where we calculated the daily average wastage amount (multiply per session wastage and daily mean session number), as well as the average of all video sessions for other metrics. The first observation is that the overall data wastage ratio across the seven algorithms ranges from 25.2% to 51.7%, which means that a quarter to half of the downloaded data is in fact wasted. In addition, data wastage amount is on average 1.17~6.17 Petabyte each day. Given the pricing of Amazon CDN [28], such amounts of data wastage can cost the streaming vendor tens to hundreds of millions of dollars each year.

Second, in the "Skip v.s. Departure" column of Table 3, we evaluated the percentage of data wastage caused by video skip versus early departure. We can see that video skip incurs about twice as much data wastage as early departure in almost all the algorithms (except for BB due to its bandwidth-limiting strategy at the beginning of each video session [7]). This is intuitive as viewers can only quit at most once in each video session while on average skip 2 ~ 3 times (c.f. Fig. 2).

Third, LBG and BBA exhibit substantial data wastage (51.7% and 50.5%) which is a result of their large buffer size and conservative bitrate adaptation logic (reflected by video bitrate). In contrast, their streaming performance measured by rebuffering duration and rebuffering frequency is much lower than others, as their higher buffer occupancy can absorb larger throughput fluctuations to prevent playback rebuffering.

Table 3
Evaluation Results of Existing Streaming Algorithms.

Streaming algorithm	Buffer size	Wastage ratio (%)	Daily mean wastage amount (Petabyte)	Skip v.s. Departure	Bitrate (Mbps)	Buffer occupancy (s)	Rebuffering duration (s)	Rebuffering frequency	QoE
LBG	184s	51.7	5.75	69% v.s. 31%	1.77	35.9	1.23	1.18	0.92
BBA	240s	50.5	6.17	63% v.s. 37%	1.31	40.7	0.80	1.29	0.94
MPC	30s	28.3	1.19	69% v.s. 31%	2.99	6.70	6.33	7.21	1.55
Stagefright	20MB	39.8	3.01	66% v.s. 34%	1.71	16.7	1.07	1.44	1.11
Pensieve	60s	25.2	1.17	67% v.s. 33%	3.22	5.73	8.94	9.9	1.73
BB	30s	38.5	1.24	51% v.s. 49%	1.32	6.79	9.08	4.54	0.71
SARA	20s	30.3	1.21	65% v.s. 35%	1.23	10.9	3.10	3.41	0.80

Table 4
Data Wastage of MPC Across Seven Trace Datasets.

	Dataset						
Metrics	#1	#2	#3	#4	#5	#6	#7
Wastage Ratio (%)	33.2	32.7	26.3	25.6	22.9	39.5	24.3
Wastage Amount (PB)	2.01	1.95	1.04	1.14	0.78	4.10	1.07

In comparison, although Stagefright also has a conservative bitrate adaptation logic, its data wastage (39.8%) is much lower than LBG and BBA due to its smaller buffer size (20MB or approximately 90s of video data). SARA has the smallest buffer size (i.e., 20s) among all the evaluated streaming algorithms thus achieves lower data wastage (30.3%) than Stagefright. However, such a small buffer leads to much more rebuffering events for SARA, decreasing its QoE performance.

Interestingly, although MPC and Pensieve are non-wastage-aware, they can also achieve comparatively lower data wastage (28.3% for MPC and 25.2% for Pensieve). This is due to their aggressive bitrate adaptation logics, which can result in relatively low buffer level. In comparison, while BB's strategy (i.e., restricting bandwidth) is also effective in reducing data wastage, it significantly increases the number of rebuffering events (the average rebuffering duration is 9.08, which is the largest among the seven algorithms) and thus degrades QoE (mean QoE is 0.71, which is the worst among the seven algorithms).

Table 4 compares the data wastage ratio/amount across the throughput trace dataset #1~#7. Note that we only listed the results of MPC, as results for other streaming algorithms are similar (see Appendix A.3 for full set of results). Interestingly, we found that dataset #1, #2, and #6 exhibit far more data wastage than others. Given the trace data statistics in Table 1, it appears that data wastage is more severe in networks with higher mean throughput. To further investigate this, we divided all video sessions into 10 throughput levels, with level l=0,1,…,8 collecting sessions with mean throughput within (l, l+1] Mbps, plus level 9 with mean throughput $\geqslant$9Mbps, and then summarized their wastage ratio/amount in Table 5 (full results are in available Appendix A.3).

Table 5
Data Wastage of MPC Across 10 Throughput Levels.

	Throughput Level				
Metrics	0~1	2~3	4~5	6~7	8~9
Wastage Ratio (%)	21.0	26.9	29.9	34.1	41.4
Wastage Amount (PB)	0.71	1.12	2.08	3.02	4.54

The results strongly suggest that data wastage increases as throughput level increases. This is due to the fact that high throughput levels indicate that the network is in well-covered mobile cells, non-peak hours, etc., and thus has a relatively stable network condition. In this case, the video player accumulates large amounts of buffered data more frequently, resulting in more data wastage.

3.3 Discussions

We gained two insights from the above results. First, data wastage is directly attributed to the buffered video data, as all the data in the buffer will be discarded upon early departure or video skip. However, video buffering is essential for preventing rebuffering and maintaining high QoE. Therefore, the need for reducing data wastage inherently conflicts with the high QoE requirements. One potential solution is to investigate whether a feasible tradeoff exists between data wastage and QoE, and if so, how to achieve the desired tradeoff. From Table 3, we found out two factors that can affect both data wastage and QoE, namely *buffer size* and *bitrate adaptation aggressiveness*, so exploiting these two factors could offer a solution to achieve the desired wastage-QoE tradeoffs.

Second, Table 4 and Table 5 reveal another interesting property – data wastage is not uniform but throughput-dependent. However, existing streaming algorithms were almost all designed to be one-size-fits-all, i.e., using fixed streaming parameter values (e.g., buffer size) irrespective of the network environments (e.g., ranging from 3G networks with a few Mbps mean bandwidth to 4G networks with 100+ Mbps peak bandwidth). Therefore, a major challenge for this work is to optimize the streaming parameters according to the specific network conditions so that data wastage can be controlled equally well on networks with different bandwidth capacities. In next section, we developed a new PSWA framework to tackle the above-mentioned challenges.

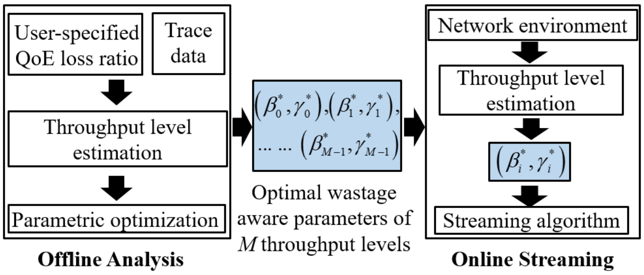


Fig. 3. The architecture of PSWA framework.

4 Wastage-Aware Video Streaming

In this section, we propose Post-Streaming Wastage Analysis (PSWA) framework. We first develop wastage-aware parameters to convert existing adaptive streaming algorithms into wastage-aware and then apply post-streaming analysis [31] to optimize the wastage-aware algorithms.

4.1 Data Wastage Awareness

Most of the existing streaming algorithms were not designed to incorporate the impact of data wastage. To this end, we design two generic wastage-aware parameters, namely *buffer limit* β and *adaptation multiplier* γ, to convert them to wastage-aware versions.

Buffer limit β. From Section 3, we found that data wastage is highly correlated with the amount of buffered video data. This suggests that limiting the buffer can control wastage. Most existing streaming algorithm originally has a buffer size (c.f. Table 3), denoted by B, but this size is typically fixed for a given algorithm and cannot be dynamically tuned based on the network condition, thus results in suboptimal performance (c.f. Section 3.2). Therefore, we designed a flexible buffering mechanism by buffer limit β.

Specifically, ignoring network latency, let t_i and f_i be the starting and completion time for transferring video segment i to the client. Let b_i be the buffer occupancy at time f_i. We schedule the starting time to transmit the next video segment at t_{i+1} to limit the buffer occupancy within β:

$$t_{i+1} = \begin{cases} f_i, \text{ if } b_i < \beta \\ f_i + b_i - \beta, \textit{ otherwise} \end{cases} \tag{7}$$

where the value of β is no longer fixed, but is to be dynamically tuned within the original buffer size B, i.e., $0<\beta<B$, according to network conditions (c.f. Section 4.2).

Adaptation multiplier γ. From Section 3, we learn that bitrate selection aggressiveness also has significant impacts on data wastage. The intuition is that an appropriate increase in the bitrate adaptation aggressiveness can reduce buffer occupancy and thus decrease data wastage. To exploit this, we develop a mechanism to regulate the adaptive algorithm's bitrate selection aggressiveness. Specifically, most of the algorithms *originally* have one or more internal metrics [29-30] which are the key criterion for them to determine video bitrate (refer to [29] for the notion of "internal metric"). Therefore, we introduce adaptation multiplier γ to multiply the internal metric, thus the bitrate selection aggressiveness can be controlled by tuning γ.

Table 6
Internal Metric and Adaptation Multiplier γ of the Existing Streaming Algorithms.

Algorithm	Internal metric	Range of γ
LBG [23]	Video segment duration over segment download time	0~3
Stagefright [24]	The sliding window of throughput measurement	0~5
BBA [25]	Mapping slope between buffer occupancy and video bitrate	0~12
MPC [26]	The harmonic mean of past throughput divided by previous estimation error	0~5
Pensieve [27]	Throughput measurement vector including past 8 video segments	0~3

It's worth noting that the definition of the internal metric in the existing streaming algorithms depends on the specific design of their adaptation logic, so the definition differs across different algorithms. Table 6 summarizes the description for the internal metric of five existing adaptive streaming algorithms, and we refer the interested readers to their original studies [23-27] for the detailed definitions.

To illustrate how the adaptation multiplier γ works, we take MPC [26] as an example, of which the definition of the internal metric is reproduced below (proposed by Yin *et al.* [26]):

$$D_k = H_k / (1 + e_k) \tag{8}$$

where D_k is the estimated throughput for determining the bitrate of segment k, H_k is the harmonic mean throughput for downloading the past 5 segments (i.e., segment k–6 ~ k–1) and e_k is the previous maximum absolute estimation error. MPC mainly relies on the estimated throughput D_k to determine video bitrate [26], so we can apply the multiplier γ to D_k to control the bitrate selection aggressiveness:

$$D_k^{'} = \gamma \times D_k \tag{9}$$

where the value of γ can be tuned to change the final output, denoted by $D_k^{'}$.

Naturally, different streaming algorithms may use different internal metrics (e.g., the throughput measurement vector in Pensieve [27] as opposed to harmonic mean in MPC) but one can apply γ in a similar fashion (see Table 6) to control their bitrate selection aggressiveness.

4.2 Post-Streaming Wastage Analysis (PSWA)

We defined two wastage-aware parameters (i.e., β and γ) in Section 4.1, which can be easily applied to existing streaming algorithms. The next challenge is to find a way to determine the optimal (wastage-aware) parameter value that can effectively control data wastage while maintaining high QoE.

Although mobile networks are known to have rapid bandwidth fluctuations, they also exhibit consistent properties over longer timescales (e.g., days) so that analysis of the network conditions in past video sessions (e.g., in the past a few days) can inform the optimization of future streaming sessions [31]. Exploiting this, Liu *et al.* [31] proposed Post-Streaming Analysis that can provide predictable streaming performance in adaptive video streaming. The idea is to exploit past streaming trace data captured as a by-product of video sessions to automatically tune streaming parameters in the adaptation logic to achieve the desired streaming performance, e.g., target rebuffering probability, in future video sessions.

Drawing on the Post-Streaming Analysis principle, we developed a novel Post-Streaming Wastage Analysis (PSWA) framework to control data wastage through optimizing the wastage-aware parameters, i.e., buffer limit β, and adaptation multiplier γ. Specifically, PSWA comprises repeating cycles of two phases, namely *offline analysis* and *online streaming*, as depicted in Fig. 3. PSWA executes offline analysis periodically, e.g., daily, to compute the optimal value of β and γ for use in online streaming, e.g., the next 24 hours.

Offline Analysis. The results in Section 3 reveal that data wastage is throughput-dependent. This suggests that a single set of parameters optimized for all kinds of network conditions is likely to be sub-optimal. To tackle this challenge, we segregate network conditions into different classes according to the throughput level (c.f. Section 3) so that the wastage-aware parameters can be optimized separately to match the characteristics of different network classes. However, while the throughput level can be calculated directly in offline analysis, as the throughput trace data are given, it cannot be known before streaming the actual video session in online streaming. Therefore, we need a way to estimate the throughput level for the new video sessions.

Video players typically prefetch a number of video segments before commencing playback. The throughput in downloading the prefetch segments reflects the current network condition and thus can be used to estimate the throughput level for the new video session. Specifically, let α be the pre-configured bitrate for the first m segments during prefetch, i.e.,

$$r_{j,k} = \alpha, k = 0,1,...m-1 \tag{10}$$

where $r_{j,k}$ denotes the selected video bitrate for the k^{th} segment in session j. After segment m-1 is received, the system can then calculate the mean throughput from

$$V_j = \frac{1}{m}\sum_{k=0}^{m-1}\frac{s_{j,k}}{d_{j,k}} \tag{11}$$

where $s_{j,k}$, and $d_{j,k}$ are size and download time for segment k in the prefetch phase of session j. We then employ a linear quantization policy to map the throughput level T_j from the mean throughput V_j:

$$T_j = \min\left(\left\lfloor \frac{V_j}{\Delta} \right\rfloor, M-1\right) \tag{12}$$

where Δ is the quantization step size and M is the maximum number of the throughput level. Based on the throughput level T_j, the next step is to divide all video sessions trace data S_j, j=0,1,…,N, into M network classes:

$$C_p = \left\{ S_j \middle| T_j = \langle p \rangle, \forall j \right\},\ p = 0,1,\ldots,M-1 \tag{13}$$

where T_j is the throughput level for video session j.

PSWA then conducts *parametric optimization* to calculate the optimal wastage-aware parameters for each network class separately. Specifically, for throughput level p, PSWA executes trace-driven simulation with streaming trace data C_p to test the effectiveness of different values of wastage-aware parameter, i.e., β_p and γ_p. Note that the trace data has two types, namely *TCP throughput trace* (replicating network condition) and *viewing behavior* trace (replicating early departure and video skip), both of which are captured as a by-product of past video sessions so no extra measurements are needed.

After the simulation, PSWA records the resultant streaming performance metrics including selected bitrates, playback rebuffering, etc., to compute the overall QoE achieved in each network class, denoted by $\{Q(\beta_p, \gamma_p) \mid p$=0,1,…,$M$–1}, where $Q(.)$ is the QoE function adopted, e.g., (5). Concurrently, PSWA also records the data wastage amount, i.e., $W(.)$, in each network class, denoted by $\{W(\beta_p, \gamma_p) \mid p$=0,1,…,$M$–1}. With these two wastage-aware parameters, PSWA quantifies the relationship between QoE and data wastage (see Appendix A.2 for more details of QoE and wastage measurement).

QoE and data wastage are inherently conflicting metrics so reducing data wastage may impair QoE. However, QoE is critical to streaming services and it is likely application, service, and even user dependent so we need a mechanism for the streaming vendor to control data wastage based on their QoE preference. One possibility is to combine QoE and data wastage into a unified utility function such that the problem becomes a utility-maximization problem. However, such a utility function does not exist in the literature and it is unclear how the utility can be normalized between QoE and data wastage.

Therefore, we adopted a different approach that the system offers an interface (e.g., a configurable video player option) for the streaming vendors to specify an acceptable QoE loss ratio, denoted by δ. The purpose of δ is to allow the streaming vendors to specify their QoE preference, e.g., they can set δ to any values within 0%~100%. Note that setting δ to 0% indicates no QoE loss, in which case PSWA will maintain the resulting QoE at the same level as that achieved by the original streaming algorithms (i.e., the algorithm without wastage-aware parameters).

In the underlying design, we denote the QoE achieved by the original streaming algorithms in throughput level p as U_p, p=0,1,…M-1. PSWA then aims at minimizing the amount of data wastage and at the same time maintaining the QoE loss within δ through tuning the two wastage-aware parameters β_p and γ_p, i.e.,

$$\begin{aligned} &\min_{\beta_p,\gamma_p} W(\beta_p,\gamma_p) \\ &s.t.\ \ 1-\frac{Q(\beta_p,\gamma_p)}{U_p} \le \delta, \\ &p = 0,1,\ldots,M-1 \end{aligned} \tag{14}$$

After solving the optimization problem, PSWA obtains the optimal wastage-aware parameters for each throughput level, denoted by $\{\beta_p^*, \gamma_p^* \mid p = 0,1,...M-1\}$.

Online Streaming. After offline analysis, the optimized wastage-aware parameters will be loaded into the video player as part of streaming metadata (e.g., MPD playlist in DASH [5]). To begin a new video session, the video player first estimates the throughput level from the prefetch process, i.e., (10)~(12), and then applies the optimal wastage-aware parameters according to the throughput level to the current video session. The rest of the streaming process is unchanged. Overall, the modification needed for the appended processes is very simple so that PSWA can be readily deployed into existing streaming platforms.

4.3 Takeaway and Deployment

Takeaway. PSWA is designed to complement (rather than replace) the underlying streaming algorithms to convert them into wastage-aware versions. Thus it can be applied to the streaming platforms already in service and is compatible with the existing video streaming protocols such as DASH. The insight behind PSWA is that mobile networks exhibit consistent properties over a timescale of days so that one can analyze past video sessions' trace data to achieve predictable performance (data wastage and QoE) for future sessions [31]. Therefore, to capture the properties of the mobile network and keep detecting whether they have evolved, PSWA employs the repeated cycle of the two-phase design (c.f. Section 4.2). This guarantees that 1) the value of the wastage-aware parameters can be continuously updated, thus maintaining consistent wastage-QoE tradeoff performance even if the network condition changes significantly, and 2) the deployment of PSWA can be highly portable to accommodate the evolution of mobile streaming infrastructure.

Deployment. In applying PSWA to rate-adaptation algorithms, the computation complexity should be low as bitrate decision needs to be performed frequently online. This can be easily achieved by PSWA as most of the computations are consolidated into the offline analysis that can be executed on the server-side. For example, the CDN server of the streaming vendors can be easily extended to record the video session's trace data for offline analysis when it delivers the video data to the players over HTTP/TCP. Moreover, the optimal wastage-aware parameters can be embedded into the meta-data file of the streaming protocols (e.g., MPD in DASH) and then are sent to the video player. For online streaming, the only computation requirement is the throughput level measurement during prefetch, which is not computationally expensive and is performed only once at the beginning of each video session. To demonstrate PSWA's feasibility, we implemented PSWA into an open-source video player (dash.js [32]) and evaluated its performance (see Section 5.5).

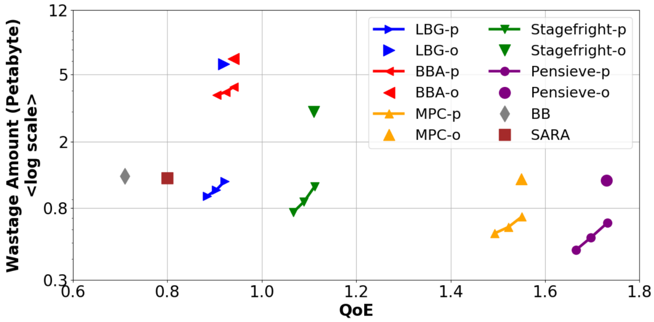


Fig. 4. Comparison of data wastage amount and QoE performance.

5 Performance Evaluation

In this section, we evaluate PSWA's effectiveness in reducing data wastage and analyze the tradeoff between data wastage and QoE.

5.1 Experiment Setup

We employed trace-driven simulations with the same setup as described in Section 3.2. PSWA was applied to optimizing the five non-wastage-aware streaming algorithms, namely LBG [23], Stagefright [24], BBA [25], MPC [26], and Pensieve [27], to turn them into wastage aware versions. In addition, the two existing wastage-aware algorithms, BB [7] and SARA [8], were evaluated to compare to the performance of PSWA.

We used a total of 60 weeks' trace data (~100,000 video sessions) in the evaluation. PSWA was configured to use the past one day's trace data in offline analysis phase to optimize the two wastage-aware parameters $\{\beta, \gamma\}$, which were then applied to online streaming phase in the next 24 hours. β is tuned within the streaming algorithm's buffer size (c.f. Table 4), and the tuning range of γ is listed in Table 6. For the throughput level, we adopted the linear mapping policy in (12) with quantization step size of Δ = 1 Mbps and M=10. Unless stated otherwise we adopted (5) as the default QoE function. The rest of the parameters are summarized in Table 3.

5.2 Performance Tradeoff

PSWA offers a tool for streaming vendors to explicitly control the tradeoff between data wastage and QoE through specifying QoE loss ratio δ. To evaluate the tradeoff trajectory, we varied δ from 0% to 4% to evaluate the tradeoff between QoE and data wastage. Fig. 7 plots the tradeoff trajectories for all seven streaming algorithms evaluated. The performance results of the original algorithms (without applying PSWA) are indicated by "-o" suffix (e.g., "LBG-o") while PSWA-optimized versions are indicated by "-p" suffix (e.g., "LBG-p").

We observed that all the five non-wastage-aware algorithms optimized by PSWA show a significant reduction in data wastage with little or even no loss of QoE. In all cases, PSWA enables them to achieve a continuous tradeoff trajectory between data wastage and QoE.

Table 7
Actual QoE Loss Proportion φ (%) versus Specified QoE Loss Ratio δ.

	QoE Loss Ratio δ (%)				
Algorithm	0	1	2	3	4
LBG	-0.15	0.96	1.75	2.86	3.93
BBA	-0.09	0.90	1.81	2.77	3.78
MPC	-0.06	0.87	1.79	2.98	3.69
Stagefright	-0.21	0.99	1.89	2.57	3.90
Pensive	-0.10	0.79	1.92	2.49	3.71

Table 8
Data Wastage Reduction Proportion ς (%) versus Specified QoE Loss Ratio δ.

	QoE Loss Ratio δ (%)				
Algorithm	0	1	2	3	4
LBG	79.9	83.2	85.4	87.2	90.2
BBA	31.6	35.7	38.2	41.8	44.4
MPC	40.3	45.1	48.7	50.4	52.3
Stagefright	64.2	68.6	71.0	74.3	74.8
Pensive	44.0	48.3	54.2	57.1	61.2

In comparison, since BB and SARA are wastage-aware algorithms, they do achieve relatively low data wastage. However, due to their one-size-fits-all model, both of them can only achieve one specific point of tradeoff and the resultant QoE is relatively low.

To evaluate PSWA's control on QoE loss, we defined a new metric φ to quantify the actual QoE loss proportion, i.e.,

$$\varphi = \sum_{\forall i}(U_i - Q_i) \Big/ \sum_{\forall i} U_i \tag{15}$$

where U_i is the QoE achieved by the original algorithm (i.e., those with "-o" suffix) for video session i, Q_i denotes the QoE achieved by the PSWA-optimized algorithms (i.e., those with "-p" suffix). We then compared φ against the specified QoE loss ratio δ in Table 7. We can see that the five algorithms performed similarly, all of which achieve the actual QoE loss proportion lower than but close to δ.

Next, we quantified the data wastage reduction by defining a new metric called data wastage reduction proportion:

$$\varsigma = \sum_{\forall i}(P_i - W_i) \Big/ \sum_{\forall i} P_i \tag{16}$$

where P_i is data wastage amount produced by the original algorithm (those with "-o" suffix) for video session i, W_i is the data wastage amount of the PSWA optimized algorithm (those with "-p" suffix). Table 8 summarizes the data wastage reduction proportion versus the specified QoE loss ratio δ. We observed that through PSWA, all the five streaming algorithms' data wastage was reduced significantly, i.e., up to 44.4%~90.2% wastage reduction within 4% QoE loss, where LBG achieves the most substantial results.

Most remarkably, PSWA manages to reduce data wastage even without any QoE loss. It is clear from the column with δ=0% in Table 8 that PSWA enables the five algorithms to achieve 31.6% to 79.9% wastage reduction.

Table 9
Data Wastage Reduction Proportion ς (%) Across Four QoE Functions (δ = 0%).

Algorithm	QoE_1	QoE_2	QoE_3	QoE_4
LBG	79.9	73.5	83.3	90.1
BBA	31.6	27.0	39.5	30.5
MPC	40.3	51.7	49.7	41.7
Stagefright	64.2	50.1	66.4	56.6
Pensive	44.0	58.2	66.1	51.2

This is due to PSWA's ability to break the one-size-fits-all model of the existing streaming algorithms. Specifically, the optimal value of a streaming algorithm's internal metric (c.f. Section 4.1) varies with the changes in network conditions [29]. However, existing streaming algorithms were commonly equipped with a fixed set of internal metrics and functioned in all kinds of networks (so-called one-size-fits-all), so the internal metrics were inevitably suboptimal that results in suboptimal QoE. By comparison, PSWA tuned γ to optimize streaming algorithms' internal metrics based on the specific network conditions and thus improve the algorithm's QoE performance beyond its original version. The potential increased QoE thus provides a QoE margin for PSWA to reduce data wastage without degrading QoE performance (see Appendix A.4 for more details).

To see if the above observations are consistent under different QoE metrics, we repeated the experiments using three more QoE functions i.e., QoE_2 ~ QoE_4 [26,33,34] (QoE_1 is defined by (5)). We set QoE loss ratio δ to 0% and summarized data wastage reduction under the four QoE functions in Table 9. We observed very similar patterns across the four QoE functions, where PSWA enables the five streaming algorithms to achieve substantial data wastage reduction without any QoE loss.

5.3 Variation Across Network Conditions

In this section, we investigate the performance variation across different network conditions. At first, we evaluated the data wastage performance over the seven throughput trace dataset #1 ~ #7, which were collected from multiple mobile operators and locations (c.f. Table 2). PSWA makes use of the past one day's trace data for offline analysis where the trace data is a combination of data from the dataset #1 ~ #7, and then unseen trace data is used for evaluating online streaming performance. Note that in this section we only show the results of MPC with setting δ to 2%, as similar results were obtained with other streaming algorithms and other settings of δ.

We summarized the results in Table 10. The observation is that PSWA enables MPC to achieve substantial data wastage reduction across all the seven datasets, ranging from 32.3% to 77.1%. Compared to MPC-o (i.e., original MPC), MPC-p (i.e., PSWA-optimized MPC) achieves more consistent wastage ratio and wastage amount across different datasets. These results strongly suggest that using the trace data with a sufficiently wide spectrum of network conditions in the offline analysis phase, PSWA can enable one algorithm to effectively control the data wastage over a wide range of network environments.

Table 10
Data Wastage of MPC Across Seven Throughput Trace Datasets (δ = 2%).

Metrics		Wastage Ratio (%)		Wastage Amount (PB)		Data Wastage Reduction ς (%)
Version		MPC-o	MPC-p	MPC-o	MPC-p	
Dataset	#1	33.2	13.9	2.01	0.90	55.1
	#2	32.7	13.6	1.95	0.91	53.2
	#3	26.3	13.4	1.04	0.61	41.3
	#4	25.6	13.1	1.14	0.67	40.7
	#5	22.9	12.9	0.87	0.59	32.3
	#6	39.5	13.1	4.10	0.96	76.9
	#7	24.3	14.0	1.07	0.63	41.1

Table 11
Data Wastage of MPC Across 10 Throughput Levels (δ = 2%).

Metrics	Algorithm	Throughput Level 0~1	2~3	4~5	6~7	8~9
Wastage Ratio (%)	MPC-o	21.0	26.9	29.9	34.1	41.4
	MPC-p	13.2	14.3	13.1	13.2	12.8
Wastage Amount (PB)	MPC-o	0.71	1.12	2.08	3.02	4.54
	MPC-p	0.48	0.65	1.05	1.17	1.39
Wastage Reduction ς (%)		32.1	41.9	49.4	61.2	69.2

To further analyze the results across different levels of throughput, we divided all video sessions into 10 throughput levels, with level l=0,1,…,8 collecting sessions with average throughput within (l, l+1] Mbps, plus level 9 with average throughput ⩾9Mbps, and then summarized their respective data wastage performance in Table 11.

We observed that through PSWA's optimization, MPC-p can effectively control data wastage across all the throughput levels. The generally higher data wastage at higher throughput levels is now compensated by PSWA with higher wastage reduction. Compared to MPC-o, MPC-p's wastage ratio is far more consistent across the 10 levels, but its wastage amount still exhibits a slight increase as the throughput level increases. We argue that this increase is inevitable as adaptive streaming algorithms typically select higher video bitrate at higher throughput levels and hence the larger video segment size would naturally lead to more data wastage. Nevertheless, after PSWA's optimization, the rising slope of wastage amount of MPC-p is much lower than that of MPC-o.

To further investigate the dynamics of PSWA with respect to throughput levels, we calculated in Table 12 the mean values of the wastage-aware parameters (i.e., β and γ) of MPC-p in different levels. There are two observations. First, the results clearly show that the optimal wastage-aware parameters vary substantially across throughput levels. This validates PSWA's throughput-level differentiation approach to optimize the parameters. Second, as throughput level increases, the buffer limit β decreases while the adaptation multiplier γ increases. This indicates that PSWA is exploiting the (better) network condition at higher throughput levels where the likelihood of low bandwidth is much lower than in lower throughput levels.

Table 12
Wastage-aware Parameters of MPC-p Across 10 Throughput Levels (δ = 2%).

Network Characters and Wastage-aware Parameters	Throughput Level 0~1	2~3	4~5	6~7	8~9
Throughput (Mbps)	0~2	2~4	4~6	6~8	≥8
Coefficient of Variation (CoV)	0.84	0.59	0.42	0.32	0.25
Buffer limit β (s)	11.2	10.3	8.8	7.9	6.9
Adaptation multiplier γ	0.8	1.3	1.7	2.1	2.6

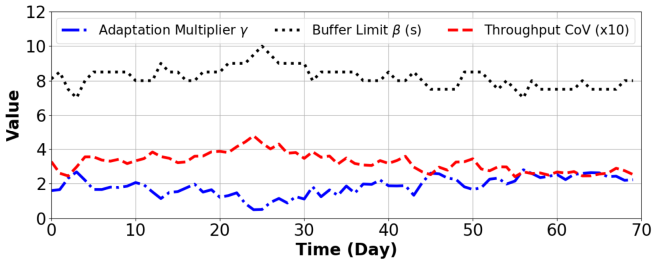


Fig. 5. The evolution of wastage-aware parameters (in throughput level 5) over a period of 70 days.

Specifically, in the higher throughput level, the smaller buffer limit directly reduces data wastage while the larger adaptation multiplier improves video quality (via more aggressive bitrate selection) and reduce data wastage (via lower buffer occupancy). PSWA thus jointly tunes β and γ to achieve more substantial data wastage reduction (see Appendix A.4 for more details).

Fig. 5 plots the daily mean values of β and γ in throughput level 5 over a period of 70 days (similar patterns can be observed in other throughput levels). Since the mean throughput at a certain throughput level is limited to a specific range (e.g., 5 Mbps ~ 6 Mbps in throughput level 5), we can ignore the impact of mean throughput but focus on the effect of throughput variations (quantified by throughput Coefficient of Variation (CoV) in Fig. 5) on the parameter value. The observation is that the values of the two parameters were constantly changing as the evolution of the throughput CoV over the 70 days. This is intuitive, for example, a network with higher throughput CoVs is more likely to cause more rebuffering events, so in this case, PSWA maintains a relatively high buffer level by tuning the parameters such that QoE degradation can be avoided. Overall, the results clearly demonstrate that through executing the offline analysis on a daily basis, PSWA was able to optimize/update the wastage-aware parameter appropriately to adapt to the changing network conditions. A deeper analysis on the two wastage-aware parameters (β and γ) is in Appendix A.4.

5.4 Sensitivity Analysis

In this section, we dissect PSWA by investigating the relative performance contribution by its key components. Specifically, we investigate the significance of: (a) tuning buffer limit β only while keeping γ to 1; (b) tuning adaptation multiplier γ only while keeping β to the algorithm's original buffer size; (c) removing the throughput level differentiation (as in an early version of this work [18])

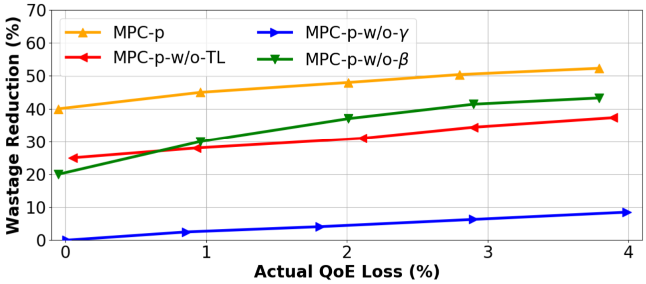


Fig. 6. Performance contributions of PSWA's key components.

We compared the performance of the full version PSWA (indicated by the "-p" suffix) to the three handicapped versions, indicated by "-p-w/o-β" (without tuning β), "-p-w/o-γ" (without tuning γ) and "-p-w/o-TL" (without differentiating throughput levels) suffixes respectively. Fig. 6 plots their performances in terms of QoE loss and data wastage reduction. Again we only showed the results for MPC as the results for other algorithms are similar.

From Fig. 6, it is clear that both the throughput level differentiation and the two wastage-aware parameters are essential to PSWA as the effectiveness of reducing data wastage drops significantly without anyone of them. In particular, the performance drops the most without tuning γ (i.e., MPC-p-w/o-γ) where the curve exhibits a more linear pattern passing through the origin.

5.5 Implementation and Real Experiments

In this section, we report results from a prototype implementation of PSWA into the well-known *dash.js* video player (version 3.11) [32] to validate PSWA's practicality and to verify its performance in real-world streaming implementations. Specifically, we first modified dash.js to support the five non-wastage-aware streaming algorithms. For Pensieve, dash.js was configured to fetch bitrate selection decisions from a specialized bitrate decision server where Pensieve's neural network is deployed. All other algorithms were embedded into "AbrController.js" of dash.js and executed directly. Next, we specified a 2% QoE loss ratio for PSWA's offline analysis and then applied the optimal wastage-aware parameters into the streaming algorithms in dash.js.

In our setup, the video server host ran Linux with the Apache httpd [35] serving video data over TCP CUBIC [36] and the video client was a Google Chrome browser running in a smartphone with Android operating system. We used an improved version of DummyNet [37] to emulate the network conditions between the client and server based on our collected TCP throughput trace data [22], along with 80 ms minimal RTT to model propagation delay. Other streaming settings (e.g., video duration, bitrate profile, etc.) were consistent with those in Section 3.2.

We ran each streaming algorithm twice, each executing 1000 video sessions (the throughput trace data was the same for both runs). Specifically, we ran streaming algorithms with their original settings (i.e., without PSWA) for the first time, and then applied the wastage-aware parameters into the algorithms to run again (i.e., with PSWA).

Table 13
Experimental Results (δ = 2%).

	Actual QoE Loss (%)	Wastage Reduction (%)
LBG	1.74	78.3
BBA	1.91	40.2
MPC	1.87	53.9
Stagefright	1.62	71.5
Pensieve	1.88	56.1

Table 13 shows the proportion of actual QoE loss and data wastage reduction for each algorithm. We observed that the actual QoE losses of the five algorithms were all within the specified QoE loss ratio 2%. Meanwhile, the data wastage reduction is significant in all cases, ranging from 40.2% to 78.3%. Overall, the experimental results verified PSWA's design goal to achieve the desired tradeoff performance between QoE and data wastage in a real-world streaming implementation. Therefore, PSWA offers an immediate and practical solution to significantly reduce data wastage in current as well as future streaming platforms.

6 Summary and Future Work

This work reveals that current video streaming systems can result in substantial data wastage due to viewer's early departure and video skip behavior. To tackle the problem, we proposed a novel PSWA framework which can reduce data wastage significantly (e.g., up to 80%) without impacting QoE. PSWA not only can convert existing on-demand adaptive streaming algorithms into wastage-aware versions, it can also be incorporated into the design of new streaming algorithms so that data wastage becomes an integrated performance metric rather than an afterthought.

This work is only the first step in this direction. There are many opportunities for future research, because data wastage is not limited to on-demand streaming platforms. For example, there is a rapid increase in live streaming services in recent years and although viewers cannot skip ahead in a live stream, their early departure would certainly result in data wastage. Similarly, the emerging 360-degree video streaming poses an even bigger challenge on data wastage due to its viewport-based streaming. More work is thus warranted to investigate these research challenges.

Acknowledgements

The authors wish to thank the associate editor and the anonymous reviewers for their insightful comments in improving this paper. This work is supported by Centre for Advances in Reliability and Safety Limited (CAiRS) under AIR@InnoHK research cluster.

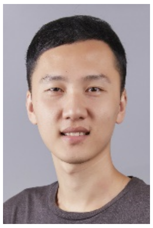
References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecase Update, 2018-2023*, Mar 2020, Cisco Inc. [Online] https://www.cisco.com/c/en/us/solutions/collateral/executive-

perspectives/annual-internet-report/white-paper-c11-741490.html
[2] A. Finamore, M. Mellia, M.M. Munafò, R. Torres, and S.G. Rao. "Youtube Everywhere: Impact of Device and Infrastructure Synergies on User Experience," *Proc. ACM SIGCOMM*, Toronto, Canada, Nov 2011, pp.345-360.
[3] F. Dobrian, V. Sekar, A. Awan, "Understanding the Impact of Video Quality on User Engagement," *ACM SIGCOMM Computer Communication Review*, Aug 2011, vol.41(4), pp.362-373.
[4] Z. Li, Q. Wu, K. Salamatian, "Video Delivery Performance of A Large-scale VoD System and the Implications on Content Delivery," *IEEE Transactions on Multimedia*, June 2015, vol.17(6), pp.880-892.
[5] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," *Proc. ACM Conf. Multimedia Syst.*, San Jose, USA, Feb 2011, pp.133-144.
[6] *Mobile Data Plan of China Mobile in Hong Kong*. [Online] https://www.hk.chinamobile.com/en/corporate_infomation/Service_Plans/
[7] L. Chen, Y.P. Zhou, and D.M. Chiu, "Smart Streaming for Online Video Services," *IEEE Trans. Multimedia*, vol.17(4), Apr 2015, pp.485–497.
[8] H. K. Yarnagula, P. Juluri, S. K. Mehr, V. Tamarapalli, and D. Medhi, "QoE for Mobile Clients with Segment-aware Rate Adaptation Algorithm (SARA) for DASH Video Streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol.15(2), June 2019.
[9] X. Chen, T. Tan, G. Cao, "Energy-Aware and Context-Aware Video Streaming on Smartphones," *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, June 2019, pp.861-870.
[10] N. Li, Y. Hu, Y. Chen, "Lyapunov Optimized Resource Management for Multiuser Mobile Video Streaming," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.29(6), June 2018, pp.1795-1805.
[11] G. Huang, W. Gong, B. Zhang, "An Online Buffer-Aware Resource Allocation Algorithm for Multiuser Mobile Video Streaming," *IEEE Transactions on Vehicular Technology*, vol.69(3), Mar 2020, pp.3357-3369.
[12] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Commun. Surveys Tuts.*, vol.17(1), Q1 2015, pp.469-492.
[13] P. Juluri, V. Tamarapalli, and D. Medhi, "Measurement of Quality of Experience of Video-on-Demand Services: A Survey," *IEEE Commun. Surveys Tuts.*, vol.18(1), Q1 2016, pp.401-418.
[14] J. Kua, G. Armitage, and P. Branch, "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP," *IEEE Commun. Surveys Tuts.*, vol.19(3), Q3 2017, pp.1842 - 1866.
[15] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP," *IEEE Commun. Surveys Tuts.*, vol.21(1), Q1 2019, pp.562-585.
[16] *Tencent Video*. [Online] https://v.qq.com/
[17] L. Chen, Y. Zhou and D. M. Chiu, "Video Browsing-A Study of User Behavior in Online Vod Services," *Proc. IEEE ICCCN*, Nassau, Bahamas, Oct 2013, pp.1-7.
[18] G. Zhang, J.Y.B. Lee, "On Data Wastage in Mobile Video Streaming," *Proc. IEEE WCNC*, Barcelona, Spain, Jun 2018, pp. 1-6.
[19] *Best Practices for Creating and Deploying HTTP Live Streaming Media for the iPhone and iPad*, Apple Inc, retrieved on Aug 2016. [Online] https://developer.apple.com/library/ios/technotes/tn2224/_index.html
[20] H. Riiser, P. Vigmostad, C. Griwodz, P. Halvorsen, "Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications," Proc. *ACM Multimedia Syst.*, Oslo, Norway, Feb 2013, pp.114-118.
[21] G. Yi, D. Yang, A. Bentaleb, "The ACM Multimedia 2019 Live Video Streaming Grand Challenge," *Proc. ACM International Conference on Multimedia*, Nice, France, Oct 2019, pp.2622-2626.
[22] *Mobile Throughput Trace Data*. [Online] http://sonar.mclab.info/tracedata/TCP/
[23] C. Liu, I. Bouazizi and M. Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," *Proc. ACM Conf. Multimedia Syst.*, San Jose, USA, Feb 2011, pp.169-174.
[24] *The Android Open Source Project*, Android Git Repositories, retrieved on Feb 2017. [Online] https://android.googlesource.com/platform/frameworks/av/+/master/media/libstagefright/httplive/LiveSession.cpp
[25] T.Y. Huang, R. Johari, N. McKeown, M. Trunnell and M. Watson, "A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," *Proc. ACM SIGCOMM*, Chicago, Illinois, USA, Aug 2014, pp.187-198.
[26] X. Yin, A. Jindal, V. Sekar and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *Proc. ACM SIGCOMM*, London, United Kingdom, Aug 2015, pp.325-338.
[27] H. Mao, R. Netravali, M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," *Proc. ACM SIGCOMM*, Los Angeles, CA, USA, Aug 2017, pp.197-210.
[28] *Amazon CDN Pricing*. [Online] https://aws.amazon.com/cloudfront/pricing/?nc1=h_ls
[29] Y. Liu and J.Y.B. Lee. "A Unified Framework for Automatic Quality-of-Experience Optimization in Mobile Video Streaming," *Proc. IEEE INFOCOM*, San Francisco, CA, USA., Apr 2016, pp.1-9.
[30] G. Zhang and J.Y.B. Lee, "Ensemble Adaptive Streaming-A New Paradigm to Generate Streaming Algorithms via Specializations," *IEEE Transactions on Mobile Computing*, vol.19(6), June 2020, pp.1346-1358.
[31] Y. Liu and J.Y.B. Lee, "Post-Streaming Rate Analysis - A New Approach to Mobile Video Streaming with Predictable Performance," *IEEE Trans. Mobile Comput.*, vol.16(12), Dec 2017, pp. 3488-3501.
[32] *dash.js*. [Online] https://github.com/Dash-Industry-Forum/dash.js/wiki.
[33] T. Hoßfeld, C. Moldovan, C. Schwartz, "To Each According to His Needs: Dimensioning Video Buffer for Specific User Profiles and Behavior," *Proc. IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, ON, Canada, Jul 2015, pp.1249-1254.
[34] X. Liu, F. Dobrian, H. Milner, J. Sekar, V. Jiang, I. Stoica and H. Zhang, "A Case for a Coordinated Internet Video Control Plane," in Proc. ACM SIGCOMM, Helsinki, Finland, Aug. 2012, pp. 359-370.
[35] *Apache HTTP Server Project*. May 2016. [Online]. http://httpd.apache.org/

[36] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-friendly High-speed TCP Variant," *ACM Operating Sys. Rev.*, vol. 42(5), July, 2008, pp. 64–74.

[37] *An Improved Version of Dummynet.* [Online] https://github.com/mclab-cuhk/netmap-ipfw



Dr. Guanghui Zhang is currently a Post-Doctoral Fellow with the Centre for Advances in Reliability and Safety Limited (CAiRS), the Hong Kong Polytechnic University, Kowloon, Hong Kong, where he participated in the research and development of multimedia technology. He received the Ph.D. degree in Information Engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2020. Before that, he obtained the Master degree in Electronic Science and Technology from Peking University, Beijing, China, in 2016.

Ke Liu received the B.Eng. and Ph.D. degrees in information engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2008 and 2013, respectively. He was a Post-doc Scholar in School of Industrial Engineering in Purdue university, IN, US, from 2017 to 2018. He is currently an associate professor in the Advanced System Group under the Key Laboratory of Computer System and Architecture at the Institute of Computing Technology, Chinese Academy of Science, Beijing, China, where he participated in the research of protocol optimization and cloud computing.

Dr. Haibo Hu is an associate professor in the Department of Electronic and Information Engineering, Hong Kong Polytechnic University and the programme leader of BSc (Hons) in Information Security. His research interests include cybersecurity, data privacy, internet of things, and adversarial machine learning. He has published over 80 research papers in refereed journals, international conferences, and book chapters. As principal investigator, he has received over 20 million HK dollars of external research grants from Hong Kong and mainland China as of year 2020. He has served in the organizing committee of many international conferences, such as ACM GIS 2021, 2020, IEEE ICDSC 2020, IEEE MDM 2019, DASFAA 2011, DaMEN 2011, 2013 and CloudDB 2011, and in the programme committee of dozens of international conferences and symposiums. He is the recipient of a number of titles and awards, including IEEE MDM 2019 Best Paper Award, WAIM Distinguished Young Lecturer, ICDE 2020 Outstanding Reviewer, VLDB 2018 Distinguished Reviewer, ACM-HK Best PhD Paper, Microsoft Imagine Cup, and GS1 Internet of Things Award.

Vaneet Aggarwal (S'08 - M'11 - SM'15) received the B.Tech. degree in 2005 from the Indian Institute of Technology, Kanpur, India, and the M.A. and Ph.D. degrees in 2007 and 2010, respectively from Princeton University, Princeton, NJ, USA, all in Electrical Engineering. He was a Senior Member of the Technical Staff Research with AT&T Labs Research, Bedminster, NJ, USA, from 2010 to 2014. He was an Adjunct Assistant Professor at Columbia University, NY from 2013-2014, and an Adjunct Professor at IISc Bangalore, India from 2018-2019. He is currently a faculty at Purdue University, West Lafayette, IN, USA. His current research interests are in communications and networking, video streaming, cloud computing, and machine learning.

Dr. Aggarwal received Princeton University's Porter Ogden Jacobus Honorific Fellowship in 2009, the AT&T Vice President Excellence Award in 2012, the AT&T Senior Vice President Excellence Award in 2014, the 2017 Jack Neubauer Memorial Award recognizing the Best Systems Paper published in the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the 2018 Infocom Workshop HotPOST Best Paper Award. He is on the Editorial Board of the IEEE TRANSACTIONS ON COMMUNICATIONS and the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING.

Jack Y. B. Lee (M'95–SM'03) received his B.Eng. and Ph.D. degrees in Information Engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 1993 and 1997, respectively. He is currently an Associate Professor with the Department of Information Engineering of the Chinese University of Hong Kong. His research group focuses on research in multimedia communications systems, mobile communications, protocols, and applications. He specializes in tackling research challenges arising from real-world systems. He works closely with the industry to uncover new research challenges and opportunities for new services and applications. Several of the systems research from his lab have been adopted and deployed by the industry.