On Data Wastage in Mobile Video Streaming

Guanghui Zhang
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong
Email: ghzhang@ie.cuhk.edu.hk

Jack Y. B. Lee
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong
Email: yblee@ie.cuhk.edu.hk

***Abstract*—Mobile video streaming is now ubiquitous among mobile users. This work investigated an often-neglected problem - data wastage where downloaded video data were not played back due to user early departure. Empirical measurements showed that data wastage is significant, e.g., around 20% of data downloaded were in fact wasted. Moreover, substantial data wastage exists not only in current commercial streaming platforms, but also in advanced adaptive streaming algorithms proposed in the literature. This work developed a new Post-Streaming Wastage Analysis (PSWA) framework to tackle the problem by converting existing adaptive streaming algorithms into wastage-aware. PSWA enables the service provider to control the tradeoff between data wastage and streaming quality-of-experience (QoE). Most remarkably, PSWA can achieve significant data wastage reduction (e.g., over 70%) even without negatively impacting QoE. PSWA can be applied to existing or future adaptive streaming algorithms and thus offers a practical solution to data wastage in current and future streaming services.**

Keywords—Video Streaming, Mobile Network, Data Wastage, Quality-of-Experience

I. INTRODUCTION

Mobile 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 videos were watched from start to finish. In fact, recent studies [2,3] as well as our own investigations revealed that a significant portion of videos were never watched completely - known as *early departure*.

A side-effect of early departure is that some of the downloaded video data will be *discarded* upon departure and the bandwidth consumed in transferring them will be wasted – we call this *data wastage* in the rest of the study. At first glance such data wastage may not appear to be a significant issue. However, mobile video streaming has practically all migrated to some forms of HTTP-based transfer protocol (e.g., HLS [4], Android [5], DASH [6]). Common to these protocols is the use of *progressive download* where video data are requested from a HTTP server which then transfers data at a rate allowed by HTTP/TCP. If the HTTP/TCP throughput is higher than the video bitrate then the client will fetch video data ahead of their playback schedules and buffer them locally. This can improve streaming performance significantly as the buffered data can be used to absorb mobile networks' bandwidth fluctuations.

However, the same fetch-ahead buffering mechanism can also increase data wastage significantly in the presence of early departure. For example, using an empirical dataset obtained from a production mobile streaming service we found that early departure can result in as much as 20% data wastage. According to a study by Cisco [1], mobile video streaming will consume 75% of all mobile data usage in 2020. This translates into a loss of 15% of *all* mobile data delivered which has far-reaching consequences.

First, many mobile data services nowadays either have a hard data cap after which additional data must be purchased, often at a much higher price, or impose a so-called fair-use policy where the bandwidth will be artificially lowered to a very low bitrate (e.g., 384 kbps even under LTE) if the usage has exceeded a given fair-use quota. A common fair-use quota is 5GB per month so the user in the above example will lose a significant portion of his/her data quota just for transferring video data which are never watched.

Second, if the mobile operator offers unlimited data plan then data wastage will not directly impact the user. Nevertheless, wasted data still consume precious bandwidth from the operator's network which otherwise can be used to deliver content to other users. Given the immense cost of the infrastructure, even a tiny percentage of wasted bandwidth can be financially significant to mobile operators.

One method to reduce wastage is to limit the client buffer size. Taking it to the extreme, if the player only buffers at most one video segment then the worst-case data wastage will only be one segment (i.e., a few seconds' worth of video data). 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 interruption, also known as rebuffering, can be reduced. Too small a buffer will likely lead to frequent playback rebuffering which can be an even bigger problem than data wastage. This is especially important in mobile networks where rapid and substantial bandwidth fluctuations are the norm rather than the exception.

The fundamental question is whether a feasible tradeoff between Quality-of-Experience (QoE) and data wastage exists in today's mobile networks and if so, how to achieve a desired tradeoff in a streaming platform. This work is the first attempt to provide an answer to these questions by applying the

978-1-5386-1734-2/18/$31.00 ©2018 IEEE

principle of *post-streaming analysis* proposed by Liu and Lee [7,8] to incorporate the impact of data wastage into the design and optimization of adaptive streaming algorithms.

We develop a new Post-Streaming Wastage Analysis (PSWA) framework where throughput trace data from past streaming sessions are analyzed to quantify the tradeoffs between data wastage and streaming QoE. The results are then used to automatically optimize streaming parameters such that the desired tradeoff between data wastage and QoE can be achieved. Our results show that while tuning the client buffer size alone can reduce data wastage substantially (e.g., 80%) with a small tradeoff in QoE (e.g., 5%), tuning *both* client buffer size and an internal parameter in the streaming algorithm can produce even better performances.

Remarkably, by jointly tuning both parameters PSWA was able to achieve significant wastage reduction (e.g., 76%) with *zero* tradeoff in QoE. PSWA can be applied to and turn any existing adaptive streaming algorithm into wastage-aware, thus offering a practical solution to significantly reduce data wastage in current and future streaming platforms.

The rest of paper is organized as follows: Section II reviews previous related works; Section III investigates data wastage in mobile streaming; Section IV presents the PSWA framework; Section V evaluates and compares the performance of PSWA; Section VI summarizes the study.

II. Related Work

Much work has been done in mobile video streaming. A comprehensive review of the area is beyond the scope of this work. We refer the interested readers to the studies by Seufert *et al.* [9], Juluri *et al.* [10] and Kua *et al.* [11] for survey and comparison of existing streaming algorithms.

Existing adaptive streaming algorithms were primarily designed to improve streaming quality. Much of the intelligence of an adaptation algorithm is in selecting the best video bitrate from the ones available at the server so that playback continuity can be maintained (or a given QoE metric optimized). Now as data wastage does *not* impact streaming QoE directly, it is no surprise that data wastage is often neglected in existing adaptive 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 users often abort playback early, resulting in data wastage of 25%-39% for PC players and 35%-48% for mobile users during peak hours. In another study, Plissonneau *et al.* [12] measured HTTP streaming traffic in an ISP and found that less than half of the videos were fully downloaded.

A key reason to the high rate of early departure is due to the inherent nature of online contents where users often explore videos from a wide range of sources and watch only those they found interesting. Chen *et al.* in their measurement study [13] found that users spent the majority of time in video browsing mode and watch an entire video only around 20% of the time. In a separate work Chen *et al.* [14] proposed a model for user watch-time distribution based on a combination of exponential distribution and power law distribution.

The above previous work all reported significant early departure behavior in mobile video services. In a recent study, Chen *et al.* [3] looked into the consequence of early departure - data wastage in a large video site (Tencent) and found that over 20% of bandwidth was wasted for delivering video data that were never watched. To address the problem, they developed a server-side Behavior-Based (henceforth called BB) streaming strategy to reduce wastage for non-adaptive video streaming. BB was designed for the scenario where the network is already fully utilized. It reduced wastage through limiting the transmission rate to 1.05 times the video bitrate (as opposed to as fast as TCP allows) during a user's browsing phase [13]. The bandwidth saved can then be reallocated to other streams to improve their QoE. Their simulation results showed that BB can reduce data wastage by 28%.

In comparison, the PSWA framework developed in this study offers three important contributions beyond the previous work. First, to the best of our knowledge, PSWA is the first solution for controlling data wastage in *adaptive* video streaming algorithms. Second, PSWA is designed to complement (as opposed to replace) adaptive streaming algorithms by turning them into wastage-aware and hence can be applied to streaming platforms already in service. Third, PSWA offers a tool for the service provider to control the tradeoff between data wastage and streaming QoE. For example, given an existing streaming algorithm a service provider can specify an acceptable tradeoff in QoE (e.g., 5%) and then PSWA will analyze the past throughput trace data to automatically tune parameters to minimize data wastage.

III. Data Wastage in Mobile Streaming

In this section, we first investigate the impact of user early departure on data wastage using a real-world dataset. Next, we employ trace-driven simulations to study the data wastage performance of five existing adaptive streaming algorithms.

A. An Empirical Study

Through collaboration with an anonymous mobile operator we were able to obtain the full packet-level capture (i.e., tcpdump) in one of their production video servers serving their mobile subscribers. The server supports adaptive streaming using Apple's HLS protocol [4]. We collected three months' data totaling over 60,000 streaming sessions. Among them 40% requests have valid User-Agent field of which 59% are from Android devices while the remaining 41% from Apple iOS devices. From the trace data we can derive the total video duration (from the m3u8 playlist), denoted by L_i, the amount/duration of video data downloaded, denoted by D_i, and the estimated viewing duration (based on the session duration), denoted by V_i, for streaming session i, $0 \le i < N$.

To quantify early departure we define the *viewing ratio* ϕ_i as the ratio of video played to the total video duration, i.e.,

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

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

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

TABLE I. DATA WASTAGE IN MOBILE VIDEO STREAMING

Video Duration	Average Viewing Ratio (%)	Average Download Ratio (%)	Average Wastage Ratio (%)	Average Wastage Amount (MB)
≤5min	53.5	86.0	34.9	8.7
5~50min	73.0	87.7	15.4	17.8
>50min	9.2	10.8	13.8	16.2
All	42.6	63.1	20.4	13.2

TABLE II. DATA WASTAGE IN EXISTING STREAMING ALGORITHMS

Streaming Algorithm	Buffer Size	Wastage Ratio (%) /Amount (MB)	Video Bitrate (Mbps)	Buffer Occupancy (s)	Rebuffering Probability (%) / Frequency*	QoE [17]
LBG	126s	18.9 / 12.8	2.45	24.0	11.0 / 1.01	2216
BBA	240s	43.7 / 28.1	1.23	91.4	0.01 / 1x10^{-4}	960
RobustMPC	30s	8.0 / 3.75	3.09	9.55	2.6 / 5x10^{-2}	2793
Stagefright	20MB	21.7 / 11.4	1.57	44.7	0.2 / 2x10^{-3}	1522
BB	30s	7.2 / 2.35	1.50	11.0	20.3 / 1.35	955

* Rebuffering frequency is the mean number of rebuffering per session.

To quantify data wastage due to early departure, we can compute the amount of data wastage in streaming session i, denoted by W_i, from the difference between total video data downloaded and estimated video data 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 actual data downloaded, segment size, full segment duration, estimated segment duration viewed for segment j respectively.

Similarly we can compute the ratio of data wastage amount for streaming session i, denoted by R_i, from

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

Table I summarizes the measurement results for the empirical dataset. First, the overall wastage ratio E[R_i|∀i] is 20.4% which is clearly not negligible. Second, short videos (0~5 mins) exhibited significantly higher data wastage ratio but significantly smaller amount of data wastage than its longer counterparts. This is because current commercial adaptive streaming algorithms usually begin a streaming session with low-bitrate version of the video and then progressively increase the bitrate afterwards if bandwidth allows. Thus the client is more likely to download and buffer more low-bitrate video data *ahead* of the playback schedule, thereby leading to more wastage in case the user departs early.

B. Trace-Driven Simulation Study

In addition to commercial streaming platforms, we employed trace-driven simulations to evaluate data wastage performance of recent adaptive streaming algorithms proposed in the literature. The simulator replicated mobile network behavior by replaying throughput trace data captured in a production mobile network over 3 months in three different locations. Video characteristics (i.e., duration) and user departure behavior are derived from the same trace data as described in Section III-A. The available video bitrates for adaptive algorithms followed the Apple profile [4], ranging from 200 kbps to 8600 kbps. This trace-driven simulator can offer evaluation of streaming algorithms in a realistic network environment and at the same time allow measurement of more detailed performance metrics such as playback rebuffering and QoE which are otherwise difficult, if not impossible, to achieve in a production network.

We implemented five streaming algorithms, namely LBG [15] – a bandwidth-based adaptive streaming algorithm; BBA [16] – a buffer-based adaptive streaming algorithm; Robust-MPC [17] – a hybrid bandwidth-buffer-based algorithm; Stagefright [5] – the hybrid bandwidth-buffer-based algorithm as implemented in the Android operating system, and BB (Behavior-Based) [3] – a server-side non-adaptive algorithm designed to reduce data wastage. Note that the client buffer size varies across different algorithms. We adopted the buffer size of 240s for BBA and 30s for Robust-MPC/BB as in their original studies. Stagefright has a maximum client buffer size not in duration but in bytes - 20MB [5]. In the case of LBG [15] the buffer size is adjusted dynamically according to the bitrate selected for each video segment. In our simulations we measured an average buffer size of 126s for LBG. In all algorithms the player prefetches 4 seconds video data at the lowest bitrate before commencing playback.

For BBA, LBG, Robust-MPC and Stagefright we replicated the adaptation algorithm as in their original studies/implementation. For BB we followed the principle in the original study [3] to select the video bitrate to be half of the estimated available bandwidth (nearest bitrate in Apple profile not higher than the calculated bitrate) by means of measuring the average throughput achieved in the previous 200 seconds (we also tested measurement duration from 20 seconds to 400 seconds which produced similar results).

In addition to data wastage we also measured four other performance metrics, namely *video bitrate* – defined as the average bitrate selected; *buffer occupancy* – defined as the average buffer level; *rebuffering probability* – defined as the proportion of sessions encountering at least one rebuffering event; *rebuffering frequency* – defined as the mean number of rebuffering events per session; and *QoE* – calculated from the QoE function proposed by Yin *et al.* [17], reproduced below:

$$Q = \frac{1}{K}\left(\sum_{k=1}^{K} r_k - \sum_{j=1}^{K-1} \left| r_{k+1} - r_k \right| - 3000 \times T_p - 3000 \times T_s \right) \quad (5)$$

where T_p is the total rebuffering duration, T_s is the startup delay, r_k is the bitrate selected for segment k and K is the total number of segments. The coefficients for T_p and T_s (i.e., 3000) follows Yin *et al.* [17]. This QoE function not only accounts for video bitrate and rebuffering, but also incorporates the impact of video quality variations due to bitrate adaptations (i.e., the second term in (5)).

Table II summarizes the simulation results. BBA exhibited the highest wastage ratio (43.7%) which is a result of its large client buffer (240s) and its relatively conservative adaptation algorithm as reflected by the lower video bitrate (1.23 Mbps) and QoE numbers. In contrast, its streaming performance as measured by rebuffering probability is the best as the large client buffer and conservative adaptation algorithm tended to buffer more video data to prevent playback buffering.

By contrast, LBG achieved substantially lower data wastage (18.9% vs 43.7%) but at the expense of significantly more rebuffering (11.0% vs 0.01%). Nevertheless its overall QoE is still higher than that of BBA as LBG's more aggressive adaptation algorithm selected higher video bitrates. As the buffer occupancy shows, LBG on average buffered far less video data than BBA which reduced data wastage in case a user departs early.

Interestingly, although Stagefright has a higher wastage ratio (21.7%) than LBG (18.9%), its average amount of data wastage per session is lower (11.4MB vs 12.8MB) due to the lower video bitrate selected by Stagefright. This shows that it is important to compare not only wastage ratio but the amount of data wastage as well. Compared to LBG, Robust-MPC exhibited even lower data wastage (8.0% vs 18.9%). Robust-MPC also achieved the highest video bitrate leading to the best QoE among all algorithms tested.

Finally, the BB algorithm was specifically designed to reduce data wastage. The results in Table II show that BB's strategy is effective in reducing data wastage (7.2%), which is the lowest among all algorithms tested. However, the low initial transmission rate significantly increased rebuffering probability at 20.3%. Note that BB was designed for non-adaptive streaming so may not be directly applicable to today's adaptive streaming platforms. Moreover it was designed for the scenario where the bottleneck network (e.g., between the content provider and mobile operator) is shared by a large number of multiple streaming sessions and this may be the reason why it does not work well in the simulated mobile network where the last leg is the bottleneck.

Overall the results demonstrated that different adaptation algorithms were designed with different tradeoffs but two factors clearly impact data wastage, namely client buffer size and adaptation aggressiveness. The challenge is to find a way to turn an adaptive streaming algorithm into wastage-aware and automatically optimize it to achieve a desired tradeoff between data wastage and streaming QoE.

IV. Wastage-Aware Video Streaming

In this section, we propose a new Post-Streaming Wastage Analysis (PSWA) framework to enable the explicit control of tradeoff between data wastage and QoE. We first develop two ways to turn existing adaptation algorithms into wastage-aware and then apply post-streaming analysis to automatically tune the wastage-aware adaptation algorithms.

A. Wastage Awareness

Except for BB [3] none of the existing streaming algorithms were designed to incorporate the impact of data wastage. Therefore the first challenge is to develop a general mechanism to turn an adaptive streaming algorithm into wastage aware while keeping their original adaptation logic intact. We exploit two insights we gained from Section III-B.

First, we found that data wastage ratio is highly correlated with the average client buffer occupancy. Intuitively the more data the client buffers the more data will be wasted when user departs early. This motivates us to introduce a mechanism to control the buffer occupancy.

Specifically, ignoring network latency, let t_i and f_i be the beginning and completion time for transferring video segment i to the client. Let b_i be the buffer occupancy at time f_i. Then to maintain a target buffer occupancy of β we can schedule the start time to transmit the next video segment at t_{i+1} given by

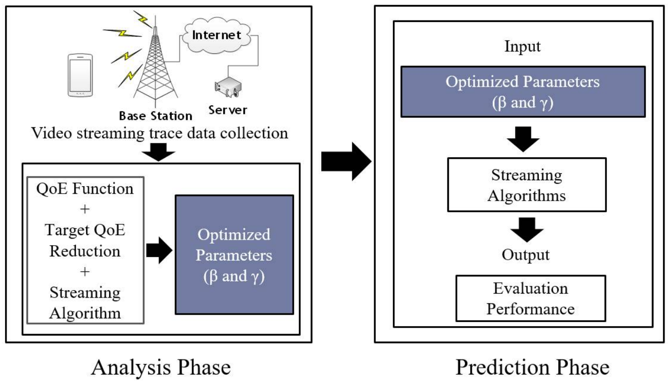

Fig. 1. The two phases in Post-Streaming Wastage Analysis.

TABLE III. Internal Parameter to be Optimized by PSWA

Algorithms	Internal Parameter	Range of γ
LBG	Video segment duration over segment fetch time [15]	0~5
BBA	Bitrate selection slope [16]	0~15
Robust-MPC	Estimated bandwidth [17]	0~5
Stagefright	Estimated bandwidth [5]	0~5

$$t_{i+1} = \begin{cases} f_i, \text{ if } b_i < \beta \\ f_i + b_i - \beta, \text{ otherwise} \end{cases} \quad (6)$$

Note that (6) is orthogonal to the adaptation logic. The latter determines the bitrate to request for the next segment and (6) controls the time to begin the transfer. If we set $\beta=\infty$ then it is the same as the original adaptation algorithm.

Second, our experiments also revealed that the aggressiveness of bitrate adaptation also affect data wastage. Most existing bitrate adaptation algorithms have one or more internal parameters which can affect their bitrate selection aggressiveness. For each of the four algorithms tested we picked a specific internal parameter (see Table III) and applied a multiplier γ to it to enable control of their bitrate selection aggressiveness. The choice of the internal parameter depends on the algorithm's design but is usually quite obvious. Interested readers are referred to the work by Liu and Lee [7].

The above modifications can be easily applied to most adaptive streaming algorithms. The next challenge is to find a way to optimize the two parameters to achieve the desired tradeoff between data wastage and QoE.

B. Post-Streaming Analysis

Post-streaming analysis [7,8] is a way to provide predictable streaming performance in adaptive video streaming. The idea is to exploit past throughput trace data captured as a by-product of streaming to automatically tune an internal parameter in the adaptation logic for future streaming sessions to achieve the desired streaming performance target, e.g., target rebuffering probability.

Drawing on the post-streaming analysis principle we developed a new Post-Streaming Wastage Analysis (PSWA) framework to use streaming trace data to automatically tune the target buffer occupancy β and the adaptation multiplier γ to control data wastage. Specifically, PSWA comprises repeating cycles of two phases, namely the analysis phase and the prediction phase as depicted in Fig. 1.

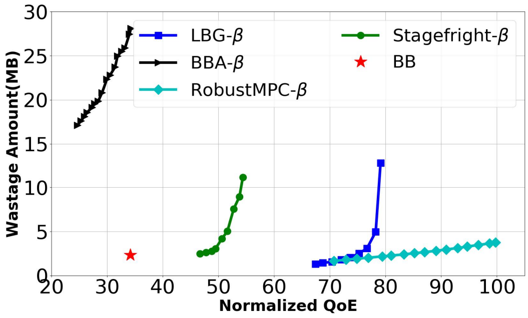

Fig. 2. Tradeoffs between data wastage and QoE (optimize buffer only).

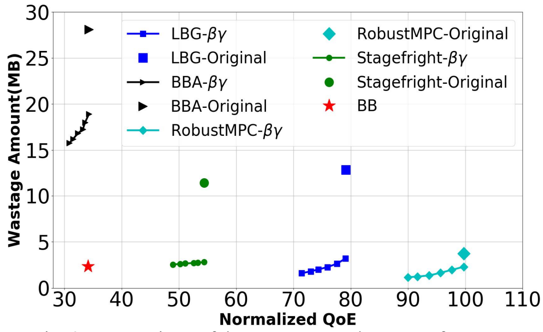

Fig. 3. Comparison of data wastage and QoE performance.

The analysis phase is executed periodically, e.g., daily, to compute the target buffer occupancy β and multiplier γ for use in the prediction phase (e.g., one day). In addition to throughput trace data captured during past streaming sessions, PSWA also records past sessions' video duration and viewing duration, both of which impact data wastage performance. PSWA then makes use of a sliding window of past streaming trace data (e.g., 7 days) to simulate streaming sessions using a range of values for β and γ respectively in the analysis phase.

PSWA records the streaming performance metrics including selected video bitrates, playback rebuffering events, etc., to compute the overall streaming QoE, denoted by $Q_i(\beta, \gamma)$, using (5) for each simulated streaming session i. Concurrently, PSWA also records the amount of data wastage for each streaming session, denoted by $W_i(\beta, \gamma)$. These metrics captured the relations between QoE and data wastage w.r.t. the target buffer occupancy β and the adaptation multiplier γ.

Not surprisingly, QoE and data wastage are conflicting metrics so we need a way for the service provider to specify the desired tradeoff between them. 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 utility function does not exist in the literature and it is unclear how the utility can be normalized between QoE and data wastage. In fact their relative significance could vary depending on type of content, data subscription plan, user preference, and so on.

Therefore we adopted a different approach whereby the service provider specifies a target QoE reduction (e.g., 5%), denoted by δ, for the purpose of reducing data wastage. Let β_o be the client buffer size in the original adaptive streaming algorithm. Then the mean QoE in the analysis phase for streaming sessions using the unmodified adaptive streaming algorithm, denoted by Q_{ori}, is given by:

$$Q_{ori} = \mathrm{E}\left[Q_i(\beta_o,1) \mid \forall i\right] \tag{7}$$

The analysis phase is then aimed at minimizing the amount of data wastage subject to the QoE reduction target δ, i.e.,

$$\min_{\beta,\gamma}\left\{\sum_{\forall i} W_i(\beta,\gamma)\right\} \quad s.t. \quad 1-\frac{E[Q_i(\beta,\gamma) \mid \forall i]}{Q_{ori}} \le \delta \tag{8}$$

The optimized parameters, denoted by $\{\beta^*, \gamma^*\}$, are then applied to all future streaming sessions in the prediction phase during which trace data will be captured as a by-product of actual streaming sessions for use in the next analysis phase and the process repeats. The insight behind PSWA is that mobile networks do exhibit consistent properties over a timescale of days so that one can make use of past trace data to optimize the streaming parameters for future streaming sessions. Through the use of a sliding window (e.g., 7 days) of past trace data in the analysis phase and a short prediction phase (e.g., 1 day) PSWA can also adapt to evolution in the mobile network infrastructure.

V. Performance Evaluation

In this section we evaluate PSWA's effectiveness in reducing data wastage for four adaptive streaming algorithms and analyze the tradeoffs between data wastage and QoE using trace-driven simulation.

A. Simulation Setting

We employed the same simulation setup as described in Section III-B for the experiments in this section. We applied PSWA to four existing adaptive streaming algorithms, namely LBG [15], BBA [16], Robust-MPC [17], and Stagefright [5], to turn them into wastage-aware. In addition, we also simulated the wastage-aware BB algorithm [3] for non-adaptive video streaming for comparison.

PSWA was configured to use the past 7 days' trace data in the analysis phase to optimize one or both of the two streaming parameters $\{\beta, \gamma\}$ for each streaming algorithm and then apply them in the prediction phase for new streaming sessions in the next 24 hours. The process then repeats and the simulation last for a total of 70 days with a total of sixty-thousand streaming sessions.

B. Tuning Target Buffer Occupancy Only

We first investigate the effectiveness of tuning only the target buffer occupancy on data wastage reduction. Fig. 2 plots the tradeoffs between QoE and amount of data wastage for the algorithms tested by varying the target QoE reduction δ. In this and subsequent figures we normalized the QoE results against the highest one achieved (by the original Robust-MPC, normalized to 100) to ease comparisons.

We observe that LBG and Stagefright showed significant reduction (80.3% and 54.9% respectively) in data wastage with a QoE reduction as small as 5%. By contrast, BBA and Robust-MPC exhibited a more linear tradeoff between the two metrics. In all cases PSWA enables one to control the tradeoff between data wastage and QoE.

In comparison, BB being a wastage-aware design did achieve relatively low data wastage. However due to its conservative transmission rate control and lack of bitrate adaptation its resultant QoE is relatively low as well.

TABLE IV. WASTAGE REDUCTION (%) UNDER DIFFERENT QOE FUNCTIONS

Algorithms	LBG	BBA	RobustMPC	Stagefright
QoE1 [17]	83.3	39.5	59.7	76.4
QoE2 [18]	93.2	77.0	71.7	89.1

C. Jointly Tuning Buffer and Streaming Parameter

One advantage of PSWA lies in its ability to jointly optimize multiple parameters of an adaptive streaming algorithm for the given objective function, i.e., (8). We repeated the experiments in Section B by applying PSWA to optimize both β and γ and plot the actual data wastage versus normalized QoE in Fig. 3. The wastage-aware version of the streaming algorithms is indicated by the "-$\beta\gamma$" suffix.

All four $\beta\gamma$-optimized adaptive streaming algorithms can achieve significant data wastage reduction compared to the original algorithms. Surprisingly, the $\beta\gamma$-optimized algorithms managed to reduce data wastage from 33% to 76% even *without* any QoE degradation. This is clear from Fig. 4 where at 0% actual QoE reduction all 4 modified adaptive streaming algorithms still achieved substantial wastage reduction.

This counter-intuitive result is due to PSWA framework's ability to optimize the internal streaming parameter via γ. As demonstrated by Liu and Lee [7] the optimal internal parameters in existing adaptive streaming algorithms often depend on the network and system configurations. Therefore by jointly optimizing γ along β PSWA can improve a streaming algorithm's QoE *beyond* its original version. The increased QoE thus provides the QoE margin for PSWA to reduce data wastage without degrading streaming QoE.

In addition to the above results, we also conducted simulations for different QoE functions, e.g., Hoßfeld, *et al.* [18] in Table IV (with target QoE reduction of 5%) which also shows substantial reduction in data wastage. Consistent wastage reduction is also observed over trace data obtained from different geographical locations and different user early-departure behaviors, confirming PSWA's performance consistency over a wide range of networks.

VI. SUMMARY AND FUTURE WORK

This work reveals that current adaptive streaming algorithms can result in substantial data wastage due to user early departure. Exploiting the principle of post-streaming analysis, the proposed PSWA framework can be used to reduce data wastage with little to no impact on streaming QoE performance. PSWA not only can turn an existing adaptive streaming algorithm into wastage-aware, it can also be applied to the design of new adaptation algorithms with wastage-control built-in for further performance optimization. On the other hand, in the presence of different subscription plans, e.g., fixed quota vs unlimited, throttled vs unthrottled, etc., the cost of data wastage can be different for different users as well. Much work needs to be done to address these and many other data-wastage-related open problems.

ACKNOWLEDGEMENTS

The authors wish to thank the anonymous reviewers for their insightful comments and suggestions in improving this paper. This work was funded by a research grant (GRF14203714) from the HKSAR Research Grant Council.

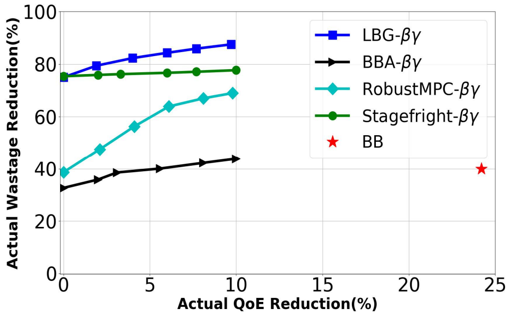

Fig. 4. Data wastage reduction versus QoE reduction.

REFERENCES

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecase Update, 2016-2021*, Mar 2017, Cisco Inc. [Online] http://www.cisco.com/c/en/us/solutions/collateral/service-provider /visual-networking-index-vni/mobile-white-paper-c11-520862.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] 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.

[4] *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

[5] *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

[6] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," *Proc. ACM Conf. Multimedia Syst.*, San Jose, USA, Feb 2011, pp.133-144.

[7] 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.

[8] 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.*, Apr 2017.

[9] 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.

[10] 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.

[11] J. Kua, G. Armitage, and P. Branch, "A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP," *IEEE Commun. Surveys Tuts.*, Mar 2017.

[12] L. Plissonneau and E. Biersack, "A Longitudinal View of HTTP Video Streaming Performance," *Proc. ACM Conf. Multimedia Syst.*, New York, NY, USA, Feb 2012, pp.203-214.

[13] 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.

[14] Y. Chen, Y. Liu, B. Zhang and W. Zhu, "On Distribution of User Movie Watching Time in a Large-Scale Video Streaming System," *Proc. IEEE ICC*, Sydney, Australia, Jun 2014, pp.1825-1830.

[15] 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.

[16] 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.

[17] 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.

[18] T. Hoßfeld, C. Moldovan, C. Schwartz, "To each according to his needs: Dimensioning video buffer for specific user profiles and behavior," *Proc. IFIP/IEEE IM*, Ottawa, ON, Canada, Jul 2015, pp.1249-1254.