Towards Reduced Latency in Adaptive Live Streaming

Yongtao Shuai
Telecommunications Lab, Saarland University
Email: shuai@nt.uni-saarland.de

Thorsten Herfet, *IEEE Senior Member*
Telecommunications Lab, Saarland University
Email: herfet@nt.uni-saarland.de

***Abstract*—Adaptive video streaming is today's predominating video delivery model in the Internet. However, existing bitrate adaptation exhibits a latency of several tens of seconds. The latency is mainly determined by the buffering delay, which aims to absorb the instantaneous mismatches between the network throughput and the video bitrate. A lot of research focuses on the adaptation algorithms, few researchers address the achievable lower bound of the latency. Our paper fills this gap by deriving an approximation of the lower bound for buffering delay. In this study, we analyze the main contributor to the latency and develop a mathematical model for the process of adaptive live streaming. Then, we present the required buffering delay given characteristics of a network (e.g., throughput and delay). This provides us with guidelines to determine a reasonable low latency for streaming systems. We validate the approximation under a trace-driven simulation of mobile network in the context of video streaming. Results show that our approximation approaches to the theoretical minimum on average.**

***Index Terms*—Internet Video, Dynamic Adaptive Streaming over HTTP (DASH), Low Latency Streaming, Bitrate Adaptation**

I. Introduction

Adaptive streaming has been widely adopted in video streaming services to improve the quality-of-experience (QoE) of video delivery. Videos are divided into small segments or chunks (typically $2\,s$ long) and encoded at multiple quality levels (typically characterized by nominal bitrates of the video), which allows streaming applications to respond to throughput fluctuations by modifying video quality.

Extensive modeling and evaluations of adaptation solutions have revealed that the state-of-the-art adaptation algorithms reach acceptable video quality only when buffering several tens of seconds [1], [2]. This leads to high playback latency in video delivery, which is undesirable especially in the context of live and interactive features with a low upper bound on the latency (a few seconds or less). The latency is mainly affected by buffering delay, which aims to absorb the instantaneous mismatches between the network throughput and the video bitrate, especially in the case of highly variable throughput; e.g., on wireless and mobile Internet paths.

Although adaptive streaming has been a very active research area, a lot of research (e.g., [1], [2], [3], [4], [5], [6]) focuses on the adaptation algorithms, few researchers (e.g., [7]) are interested in how low the latency can be. To fill this gap, our goal in this paper is to determine a reasonable lower bound on the buffering delay for video delivery. Such a low bound is absolutely necessary for the potential improvement of streaming solutions in scenarios with low latency requirements, such as the streaming of live events and augmented vision, video conferencing, and online gaming. Moreover, it provides us with baselines to design more efficient streaming systems with respect to the latency.

Our objective is to find out how low a reasonable latency for streaming systems can be. The main contributions of this paper are as follows:

- We analyze and identify the buffering delay as the key component affecting streaming latency. We develop a mathematical model of adaptive live streaming and implement a simulation testbed based on that model.
- We introduce an approximation of the lower bound of the buffering delay and conduct validation experiments in simulated scenarios with the mobile HTTP streaming throughput dataset [8]. Validation results show that our approximation is close to the theoretical minimum on average, with an underestimate error ratio of up to 0.2.

II. Towards Reduced Latency

In a live streaming system, the video content is continuously captured, encoded, and segmented while streaming. Once a video segment is available, the system can start to deliver it to the client. After the complete reception of a segment, the client decodes and renders it. In order to achieve a better responsiveness to the bitrate mismatch between network throughput and video quality, the client may buffer a specific amount of segments before the playback. In an adaptive streaming system, the video content is further encoded in several versions with respect to their video characteristics (e.g., resolutions, frame rates, and bitrates). Accordingly, the system dynamically adapts the characteristics of the video stream to varying network conditions, leading to a smoother viewing experience with less playback stalls and higher video bitrates.

A. Latency in Adaptive Live Streaming

We define the *latency* as the time difference between the time at which the content is recorded and the time at which it is displayed on a client. This latency is mainly affected by the following components: the video codec (for encoding and decoding), the delivery (traversal over the network layers at both ends) and the buffering. We assume that the video encoder will be parameterized appropriately for a given scenario to

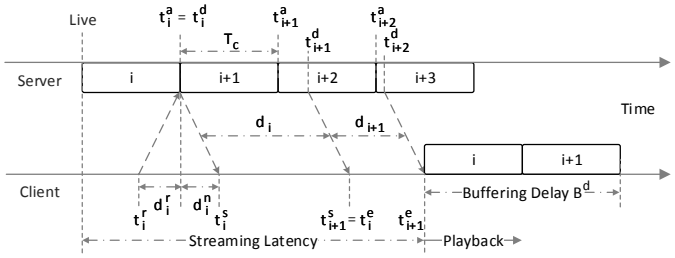


Figure 1. Streaming latency with buffering delay of two segments.

allow a sufficient remainder of the overall delay budget for the delivery and the buffering. Therefore, in this paper, we focus on the latency affected by the delivery and the buffering. The *delivery time* expresses the time required for the delivery of video data (e.g., a frame or a segment) over the network. It consists of the reception duration for a segment and the network delay along the downstream path. A client may need to store a specific amount of video data into its playback buffer before starting a video playback. We refer to this amount of video data in seconds as the *buffering delay*.

Figure 1 illustrates an example of the latency in live streaming and the time notations used as follows. We assume that the delivery of segments is performed sequentially. Consider segment i is available at time t_i^a. The client issues a request for the segment at time t_i^r and the delivery time for the request is d_i^r. Then the server starts to deliver the segment at time t_i^d. The network delay of segment i is denoted by d_i^n. At time $t_i^s = t_i^d + d_i^n$ the client starts to receive the segment and finishes the reception at time t_i^e. Thus, the reception duration for segment i can be expressed as $d_i = t_i^e - t_i^s$. Under the assumption of the sequential delivery, we have

$$t_i^d = \max\left\{t_i^a,\, t_i^r + d_i^r,\, t_{i-1}^d + d_{i-1}\right\} \ , \tag{1}$$

where $i > 1$ and $t_1^d = \max\{t_1^a,\, t_1^r + d_1^r\}$. Note that $t_{i-1}^d + d_{i-1}$ in Equation 1 indicates the time at which the server finishes sending segment $i-1$. We denote the buffering delay by B^d and the segment duration by T_c. If a client buffer contains $m \geq 1$ segments (i.e., $B^d = mT_c$) when a playback starts, the latency can be calculated as:

$$L = t_{i+m-1}^s + d_{i+m-1} - t_i^a + T_c \ , \tag{2}$$

where $i \geqslant 1, \forall j \geq 1:\ t_j^s = t_j^d + d_j^n, \forall j > 1:\ t_j^a = t_{j-1}^a + T_c = t_0 + jT_c$, $t_1^a = t_0 + T_c$, and t_0 is the start time of the content.

In adaptive streaming, adaptation algorithms strive to select segments whose video bitrates are close to the network throughput on average (i.e., $\sum_{j=0}^{m-1} d_{i+j} \sim mT_c$). So we have

$$\begin{aligned} L =& t_i^d + d_{i+m-1}^n + \sum_{j=0}^{m-1} d_{i+j} - t_i^a + T_c \\ \sim& B^d + t_i^d - t_i^a + d_{i+m-1}^n + T_c \end{aligned} \ . \tag{3}$$

Based on Equation 3, we see that the buffering delay is the biggest contributor to the latency which can be optimized under a given network scenario. It implies that given network characteristics (e.g., throughput and network delay), we can optimize m, T_c, and video bitrates (for having small d_i), in order to minimize the streaming latency. Normally, T_c and a selected set of video bitrates are given in a streaming system for preserving a high coding efficiency and video quality. So the minimization of m (the buffering delay) is the challenge in low latency adaptive streaming.

B. Adaptive Streaming Model

Consider a video as a set of N consecutive segments, each of which represents T_c seconds of the video and is encoded at different nominal bitrates in set $\mathcal{R}$. A streaming client receives the video segments and stores them into a playback buffer. Let $b(t)$ be the buffer level (in seconds) at time t.

Let t_i^s and t_i^e be the time at which the client starts and finishes the reception of segment $i \geq 1$, respectively. If the size (in $kbit$) of segment i encoded at the nominal bitrate R is given and denoted by S_i^R, or if the average bitrate of segment i is given and denoted by r_i, we have

$$S_i^R = \int_{t_i^s}^{t_i^e} C(t)\, dt \quad \text{or} \quad r_i \cdot T_c = \int_{t_i^s}^{t_i^e} C(t)\, dt \ , \tag{4}$$

where $C(t)$ is the available network throughput at time t. After the entire reception of segment i, the client may wait for Δt_{i+1} seconds and starts to receive segment $i+1$ at time t_{i+1}^s. As described in Section II-A, the reception duration of segment i is defined as

$$d_i = t_i^e - t_i^s \tag{5}$$

and the start time of receiving segment $i+1$ is given by

$$t_{i+1}^s = t_{i+1}^d + d_{i+1}^n \ . \tag{6}$$

According to Equation 1, the delivery of segment $i+1$ starts at time

$$t_{i+1}^d = \max\left\{t_{i+1}^a,\, t_{i+1}^r + d_{i+1}^r,\, t_i^d + d_i\right\} \ . \tag{7}$$

Therefore the reception delay between segment i and segment $i+1$ (i.e., the time that elapses from the moment the client finishes receiving segment i until the client starts to receive segment $i+1$) can be computed as

$$\Delta t_{i+1} = t_{i+1}^s - t_i^e \ . \tag{8}$$

As the segments are being filled into the buffer during the receiving process and being drained out for video playback, the buffer dynamics are expressed as follows:

$$b_{i+1}^s = \left((b_i^s - d_i)_+ + T_c - \Delta t_{i+1}\right)_+ \ , \tag{9}$$

where $(x)_+ = \max\{x, 0\}$, $b_i^s = b(t_i^s)$ indicates the buffer level when the client starts to receive segment i, and the initial buffer level b_1^s is assumed to be zero. Equation 9 assumes that each segment must be received in its entirety before it can be played back. Note that playback stalls events occur if $b_i^s < d_i$ or $b_i^s + T_c < d_i + \Delta t_{i+1}$. Note that our six equations are indeed linear independent and the equation system can be solved. The dynamics of the model are therefore deterministic.

In contrast to the model defined in [6], our model introduces the available time and the request-response time for live video segments into the reception delay, and relaxes the size limit of a playback buffer which may incur the waiting time for the reception due to a full buffer.

C. Minimum Buffering Delay

In the following we will address the question: what is a reasonable buffering delay for adaptive streaming if the Quality-of-Service (QoS) of a network is given. Most Internet Service Providers (ISPs) ensure the lowest available throughput (also known as *Guaranteed Throughput*) for their customers. However, they cannot ensure such network QoS anytime, and so the throughput may drop below the guaranteed throughput in a time period, e.g., due to the failure of hardware or quality degradation of network path (e.g., wireless path). Here, we define an event as *network degradation* in which the average network throughput over each segment duration is lower than the guaranteed throughput.

Given a set of video bitrates $\mathcal{R}$, the guaranteed throughput of the network is at least $\min\{\mathcal{R}\}$, such that it can support the minimum performance of video streaming service. Consider a streaming service experiences a network degradation with a duration D^{deg}. During this time period, the service cannot offer a proper video bitrate to maintain the amount of buffered data over a specific level, because the throughput is smaller than $\min\{\mathcal{R}\}$. It means that $d_i > T_c$ holds for each received segment within D^{deg} and then the buffer level decreases constantly during playback. If the buffer is empty, the client suffers from playback stalls. Therefore, a client needs to collect sufficient video data in the buffer before a network degradation occurs. Namely, there exists a minimum buffer level a client has when the network degradation starts, such that the client can avoid playback stalls during a network degradation with a duration of D^{deg}. This minimum buffer level (in seconds) denoted by B_{min}^{deg} can be calculated as:

$$\begin{aligned} B_{min}^{deg} =& K \cdot \left(\Delta t + \frac{r}{C} T_c\right) - (K-1)_+ \cdot T_c + \\ & \left(\max\left\{D_{res}^{deg},\, \Delta t\right\} + y - T_c|_{K>0}\right)_+ \end{aligned}, \tag{10}$$

where K is the number of completely received segments during the time period D^{deg} and is defined as

$$K = \left\lfloor \frac{D^{deg}}{\Delta t + \frac{r}{C} T_c} \right\rfloor , \tag{11}$$

$r = \min\{\mathcal{R}\}$ is the selected video bitrate, Δt is the reception delay of two consecutive segments, T_c is the segment duration, C is the average throughput within D^{deg},

$$D_{res}^{deg} = D^{deg} - K \cdot \left(\Delta t + \frac{r}{C} T_c\right) \tag{12}$$

denotes the remaining time of D^{deg} after the entire reception of all K segments within D^{deg},

$$y = \frac{r \cdot T_c - C \cdot \left(D_{res}^{deg} - \Delta t\right)_+}{C'} \tag{13}$$

is the time required to receive the portion of a segment (or the entire segment) after the network degradation, C' is the average throughput succeeding the network degradation, and

$$T_c|_{K>0} = \begin{cases} T_c & K > 0 \\ 0 & else \end{cases} . \tag{14}$$

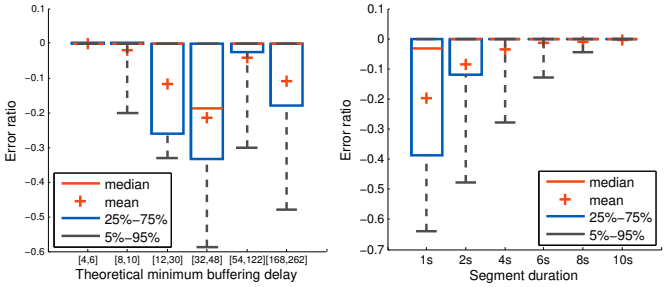
Equation 10 presents the buffer level requirement for continuous playback during a network degradation. This means that such a network with D^{deg} requires streaming services to buffer video data of at least B_{min}^{deg} prior to the network degradation. Therefore, streaming services should have a buffering delay of at least B_{min}^{deg} on a network with the network degradation of the duration D^{deg}, because a buffer level of b implies a buffering delay of at least b for streaming services.

It should be noted that Equation 10 is derived based on the following assumptions: i.) The reception of a segment is completed at the time when a network degradation starts; ii.) The reception delay of two consecutive segments is constant and equal to Δt; iii.) The average throughput for receiving each segment during the network degradation is constant and equal to C; iv.) C' fulfills $\Delta t + \frac{r \cdot T_c}{C'} \leq T_c$ such that no playback stalls occur succeeding the network degradation. As a result, our equation provides an approximation of streaming requirement with respect to the buffer level, video bitrates, the segment duration, the duration of the network degradation, and the average throughput as well as the reception delay. Note that the derivation of the exact bound requires complete throughput and delay information of the network during the streaming session. However, this is unrealistic in practice, because the perfect future network conditions are unknown. In contrast, the calculation of our approximation is straightforward. It may therefore serve as the foundation for potential requirements and improvements of streaming systems. The derivation of the equation is in the Appendix.

III. Validation

We implement a simulation testbed based on the model of adaptive streaming using Matlab. Given throughput and delay traces of the network, the testbed can simulate the process of video streaming and the buffer dynamics at the client. To validate the approximation of the minimum buffering delay, we find the theoretical minimum for a streaming session and compare it to the approximate minimum using the error ratio. The error ratio is computed as $(Approximate - Theoretical)/Theoretical$. The minimum buffering delay is rounded up to the next multiple of the segment duration T_c, since adaptive streaming is a segment-level process.

Theoretical minimum. Our theoretical minimum of the buffering delay for a streaming session is obtained in a full-search fashion. We implement a bitrate algorithm which always selects segments with $r_i = \min\{\mathcal{R}\}$ and receives two consecutive segments sequentially without the reception delay. We run the algorithm with a buffering delay of T_c under the testbed and repeat the run with an increased buffering delay by T_c until no any playback stalls occur during the streaming



a.) Segment duration of 2 s b.) Various segment durations

Figure 2. Error ratio of the approximate and theoretical minimum buffering delay, indicated with different percentiles and the mean. a.) Error ratio for scenarios with segment duration 2 s. Results are grouped into six sets according to the theoretical minimum. The range for each set is presented via closed intervals under the axis. b.) Error ratio for scenarios with segment durations of 1 s, 2 s, 4 s, 6 s, 8 s, and 10 s.

session. The buffering delay of the first run without playback stalls is the theoretical minimum.

Approximate minimum. Giving a throughput trace, we determine the duration and the average throughput of all network degradation events. We calculate the minimum buffering delay for each event based on Equation 10 and choose the maximum one as the minimum buffering delay for the streaming session. To achieve the extreme, we set $\Delta t = 0$ and $C' = \min\{\mathcal{R}\}$.

Video profiles. We use the *Big Buck Bunny* test sequence with a segment duration of 2 s [9]. The video is encoded with the following nominal bitrates using constant bitrate encoding: $\mathcal{R} = \{500\,kbps, 700\,kbps, 1200\,kbps, 3000\,kbps, 5000\,kbps\}$. This is consistent with the encoding setting for YouTube video and the bitrate range for $240p$, $360p$, $480p$, $720p$, and $1080p$, respectively[1]. We use different segment durations of 1-10 s to evaluate the validity of the approximation.

Network profiles. Riiser *et al.* [8] provide a set of video streaming throughput samples measured every one second in a mobile network. The dataset covers different vehicular mobility scenarios (metro, tram, train, bus, car, and ferry). We randomly pick 64 throughput traces from the 84 traces, and ensure that the selected traces cover all mobility scenarios and each scenario contains at least five traces[2]. Since the dataset does not include network delay measurements, we assume a one-way delay of $50\,ms$ on the network (i.e., $d_i^n = d_i^r = 50\,ms$), according to the median RTT of $100\,ms$ observed empirically in prior work [10].

Validity of approximations. The error ratios of our proposed approximations against the theoretical ones are shown via different percentiles and the mean in Figure 2. The results for scenarios with 2 s segment duration are grouped into six sets with roughly equal sizes. The ranges of minimum buffering delays in each set are presented with closed intervals: [4, 6], [8, 10], [12, 30], [32, 48], [54, 122], and [168, 262]. The numbers of results in each set are 10, 10, 11, 11, 11, and 11, respectively. Figure 2.a shows that our approximation of minimum buffering delay has median error ratios of 0 in most cases. One of exceptions is that the median and mean error ratios reach -0.19 and -0.22 in the minimum buffering delays from 32 s to 48 s, respectively. The error ratios of the approximation are from -0.02 to -0.12 on average in the case of the minimum buffering delays of [8, 10], [12, 30], [54, 122], and [168, 262]. This means that the approximate estimates are smaller than the theoretical values. We observe that the buffer level prior to the network degradation is smaller than the required buffer level B_{max}^{deg} in those cases. This is mainly due to the fact that the interval of two consecutive network degradation events is too short. In this case, the client cannot collect sufficient video data before the next network degradation to avoid playback stalls. This violates the assumption of our approximation. Nevertheless, the mean error ratio of the approximation for scenarios of 2 s segment duration is -0.08 as shown in Figure 2.b. It implies that our approximation is close to the theoretical value in most cases.

Figure 2.b shows the error ratio of the approximation under scenarios of various segment durations: 1 s, 2 s, 4 s, 6 s, 8 s, and 10 s. Consistent with the previous results, our approximation has a median error ratio of 0 for scenarios with $\geq$ 2 s segment duration. Along the increase of the segment duration, the error ratio monotonously approaches from -0.2 to 0 on average and the results converge to the mean. It implies that our approximation achieves better accuracy with larger segment durations. The reason is that the client has more time to receive data during the network degradation, because a segment in the buffer with a larger duration enables the client to playback for a longer time.

IV. Conclusion

We introduce an approximation to determine a reasonable low latency for streaming systems. We validate the approximation using a simulation methodology, which combines a bitrate-adaptive streaming simulation with a trace-driven simulation of a mobile network in the context of video streaming. Simulation results show that our approximation approaches to the theoretical minimum on average in most cases, with an underestimate error ratio of 0.2 on average in worst cases.

V. Acknowledgments

This work was supported by the Spitzencluster-Initiative from the Federal Ministry of Education and Research (BMBF) under the project number 01IC12S01X and the Intel Visual Computing Institute under the project Scalable Adaptive Low-Latency Streaming.

Appendix

Derivation of Equation 10

We derive the minimum buffer level B_{min}^{deg} based on the number of the segments k which are completely received during a network degradation with a duration D^{deg}.

$k = 0$: No video segment can be completely received during D^{deg}. The time required for the reception of a complete segment within D^{deg} is $\Delta t + \frac{r}{C}T_c$, and $k = 0$ is the case

[1]Live encoder settings, bitrates, and resolutions: https://support.google.com/youtube/answer/2853702?hl=en

[2]One of six scenarios in the dataset consists of only five traces.

that this time is bigger than D^{deg}, i.e., $D^{deg} < \Delta t + \frac{r}{C}T_c$. However, for continuous playback, the video in the buffer need to last until the reception of at least one segment completes. Therefore, the reception finishes after the network degradation. Let y be the time required to receive the portion of a segment (or the entire segment) after the network degradation. We have

$$\left(D^{deg} - \Delta t\right)_+ \cdot C + y \cdot C' = r \cdot T_c \,. \tag{15}$$

$\left(D^{deg} - \Delta t\right)_+$takes the case into account, in which the reception of the segment starts after the network degradation if $D^{deg} \leq \Delta t$ (i.e., y is the time for receiving the entire segment after the network degradation.). Then the video required in the buffer for continuous playback prior to the network degradation starts can be expressed as

$$\begin{aligned} B_{min}^{deg} =& \max\left\{D^{deg},\, \Delta t\right\} + y \\ =& \max\left\{D^{deg},\, \Delta t\right\} + \frac{r \cdot T_c - C \cdot \left(D^{deg} - \Delta t\right)_+}{C'}, , \\ & D^{deg} < \Delta t + \frac{r}{C}T_c \end{aligned} \tag{16}$$

where the term $\max\{\cdot\}$ considers the case of $D^{deg} \leq \Delta t$.

$k = 1$: Only one segment is completely received within D^{deg}. So we have $\Delta t + \frac{r}{C}T_c \leq D^{deg} < 2 \cdot \left(\Delta t + \frac{r}{C}T_c\right)$. Note that the client may initiate a request to receive an additional segment within D^{deg} but completes the reception after the network degradation. Let

$$D_{res}^{deg} = D^{deg} - k \cdot \left(\Delta t + \frac{r}{C}T_c\right) \tag{17}$$

denote the remaining time of D^{deg} after the entire reception of all k segments during the time period D^{deg}. Similar to the case of $k = 0$, the time y required to finish the reception of the additional segment after the the network degradation fulfills

$$\left(D_{res}^{deg} - \Delta t\right)_+ \cdot C + y \cdot C' = r \cdot T_c \,, \tag{18}$$

and can be calculated as

$$y = \frac{r \cdot T_c - C \cdot \left(D_{res}^{deg} - \Delta t\right)_+}{C'} \,. \tag{19}$$

Then we can compute the minimum buffer level as

$$\begin{aligned} B_{min}^{deg} =& \Delta t + \frac{r}{C}T_c + \left(\max\left\{D_{res}^{deg},\, \Delta t\right\} + y - T_c\right)_+ , \\ & \Delta t + \frac{r}{C}T_c \leq D^{deg} < 2 \cdot \left(\Delta t + \frac{r}{C}T_c\right) \end{aligned} \,, \tag{20}$$

where $D_{res}^{deg} = D^{deg} - \left(\Delta t + \frac{r}{C}T_c\right)$. The playback time of video data in the buffer should cover the time required for receiving all segments, including ones whose receptions complete within D^{deg} and the potential one whose reception finishes after the network degradation. Note that video data in the buffer increases by one segment duration T_c after the complete reception of a segment and the increase compensates the time required for receiving the following segments. Therefore, the computation of B_{min}^{deg} in Equation 20 subtracts T_c from the time required for the following reception $\max\left\{D_{res}^{deg},\, \Delta t\right\} + y$. The term $(\cdot)_+$ ensures that B_{min}^{deg} is at least $\Delta t + \frac{r}{C}T_c$, such that the client can receive one entire segment and fill the buffer with T_c seconds video data, if $\max\left\{D_{res}^{deg},\, \Delta t\right\} + y < T_c$ e.g., due to a sufficient large C'.

$k = 2$: The duration of the network degradation needs to be $2 \cdot \left(\Delta t + \frac{r}{C}T_c\right) \leq D^{deg} < 3 \cdot \left(\Delta t + \frac{r}{C}T_c\right)$, such that the client can complete the reception of exactly two segments within D^{deg}. Analog to the case of $k = 1$, we have Equation 18 and therefore obtain Equation 19, but where $D_{res}^{deg} = D^{deg} - 2 \cdot \left(\Delta t + \frac{r}{C}T_c\right)$. Because the increase of the buffer level (by T_c) due to the complete reception of a segment (i.e., the first segment in this case) compensates the decrease of the buffer level (by $\Delta t + \frac{r}{C}T_c$) during the reception of the next segment (i.e., the second segment). The minimum buffer level prior to the network degradation is

$$\begin{aligned} B_{min}^{deg} =& \Delta t + \frac{r}{C}T_c + \Delta t + \frac{r}{C}T_c - T_c + \\ & \left(\max\left\{D_{res}^{deg},\, \Delta t\right\} + y - T_c\right)_+ \\ =& 2 \cdot \left(\Delta t + \frac{r}{C}T_c\right) - T_c + \\ & \left(\max\left\{D_{res}^{deg},\, \Delta t\right\} + y - T_c\right)_+ , \\ & 2 \cdot \left(\Delta t + \frac{r}{C}T_c\right) \leq D^{deg} < 3 \cdot \left(\Delta t + \frac{r}{C}T_c\right) \end{aligned} \,. \tag{21}$$

$k = K$: Because exactly K segments are completely received within D^{deg}, it holds that

$$K \cdot \left(\Delta t + \frac{r}{C}T_c\right) \leq D^{deg} < (K+1)\left(\Delta t + \frac{r}{C}T_c\right) \,. \tag{22}$$

Similarly, we get Equation 19, where $D_{res}^{deg} = D^{deg} - K \cdot \left(\Delta t + \frac{r}{C}T_c\right)$. Accumulating the increase and the decrease of the buffer level due to the reception of all K segments, we obtain the minimum buffer level as

$$\begin{aligned} B_{min}^{deg} =& \Delta t + \frac{r}{C}T_c + \underbrace{\Delta t + \frac{r}{C}T_c - T_c + \cdots}_{K-1} + \\ & \left(\max\left\{D_{res}^{deg},\, \Delta t\right\} + y - T_c\right)_+ \\ =& K \cdot \left(\Delta t + \frac{r}{C}T_c\right) - (K-1) \cdot T_c + \\ & \left(\max\left\{D_{res}^{deg},\, \Delta t\right\} + y - T_c\right)_+ , \\ & K \cdot \left(\Delta t + \frac{r}{C}T_c\right) \leq D^{deg} < (K+1)\left(\Delta t + \frac{r}{C}T_c\right) \end{aligned} \,. \tag{23}$$

Now, we derive the number of the received segments K within D^{deg} based on the inequalities in Equation 22. We can formulate K as follows based on $D^{deg} - K \cdot \left(\Delta t + \frac{r}{C}T_c\right) \geq 0$ and $D^{deg} - (K+1)\left(\Delta t + \frac{r}{C}T_c\right) < 0$, respectively:

$$K \leq \frac{D^{deg}}{\Delta t + \frac{r}{C}T_c} \,, \tag{24}$$

$$K > \frac{D^{deg}}{\Delta t + \frac{r}{C}T} - 1 \,. \tag{25}$$

Because K is a natural number, combining Inequation 24 and Inequation 25 we have

$$K = \left\lfloor \frac{D^{deg}}{\Delta t + \frac{r}{C} T_c} \right\rfloor \ . \tag{26}$$

Considering the above cases, we can formulate the minimum buffer level required for continuous playback in case of the network degradation with a duration D^{deg} as

$$\begin{aligned} B_{min}^{deg} = & K \cdot \left(\Delta t + \frac{r}{C} T_c \right) - (K-1)_+ \cdot T_c + \\ & \left(\max \left\{ D_{res}^{deg},\, \Delta t \right\} + y - T_c|_{K>0} \right)_+ \end{aligned} \ , \tag{27}$$

where

$$K = \left\lfloor \frac{D^{deg}}{\Delta t + \frac{r}{C} T_c} \right\rfloor \ ,$$

$$D_{res}^{deg} = D^{deg} - K \cdot \left(\Delta t + \frac{r}{C} T_c \right) \ ,$$

$$y = \frac{r \cdot T_c - C \cdot \left(D_{res}^{deg} - \Delta t \right)_+}{C'} \ ,$$

and

$$T_c|_{K>0} = \begin{cases} T_c & K > 0 \\ 0 & else \end{cases} \ .$$

■

REFERENCES

[1] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro. An evaluation of bitrate adaptation methods for HTTP live streaming. *IEEE Journal on Selected Areas in Communications*, 32(4):693–705, 2014.

[2] T. Karagkioules, C. Concolato, D. Tsilimantos, and S. Valentin. A comparative case study of http adaptive streaming algorithms in mobile networks. In *ACM NOSSDAV*, 2017.

[3] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking*, 22(1):326–340, 2014.

[4] T. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. A buffer-based approach to rate adaptation: evidence from a large video streaming service. In *ACM SIGCOMM*, pages 187–198, 2014.

[5] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. Bola: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM*, 2016.

[6] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *ACM SIGCOMM*, 2015.

[7] T. Lohmar, T. Einarsson, P. Frojdh, F. Gabin, and M. Kampmann. Dynamic adaptive http streaming of live content. In *IEEE WOWMOM*, 2011.

[8] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen. Commute path bandwidth traces from 3g networks: Analysis and applications. In *ACM MMSys*, 2013.

[9] S. Lederer, C. Müller, and C. Timmerer. Dynamic adaptive streaming over http dataset. In *ACM MMSys*, 2012.

[10] P. Romirer-Maierhofer, F. Ricciato, A. D'Alconzo, R. Franzan, and W. Karner. Network-wide measurements of tcp rtt in 3g. In *TMA*, Berlin, Heidelberg, 2009. Springer-Verlag.