Stochastic Guarantees for Rate-Adaptive Streaming

Mark Akselrod, Markus Fidler
Institute of Communications Technology
Leibniz Universität Hannover

Ralf Lübben
Robert Bosch GmbH
Hildesheim, Germany

Abstract—**A major challenge in mobile video streaming applications is the variability of the wireless channel. Techniques like Apple's HLS and MPEG-DASH typically offer the video in chunks where each chunk is made available with different levels of quality and bitrate. Adaptive clients use estimates of the past network conditions to select the source bitrate of the next chunk accordingly. Frequently, this selection is based on the average performance and does not consider the variability of the channel. In this work, we present a method for bitrate adaptation that uses the distribution of the past throughput to provide statistical performance guarantees obtained by the stochastic network calculus. We show how to select the bitrate so that a certain, small probability of buffer underflow is not exceeded. Theoretical results are derived for the general case of an arbitrary throughput distribution. Numerical results are included for the example of a Gaussian distribution. Additionally, we present simulation results that show the relevant performance metrics like average bitrate and buffer level of a system that adjusts its bitrate as suggested by our method. Generally, the method presented in this paper can achieve a bitrate that is close to the average available throughput with little variability of the source bitrate, while limiting the probability of buffer underflow to a predefined level.**

I. Introduction

In recent years, online video streaming has become one of the most important Internet applications, currently accounting for the majority of all Internet traffic [1]. According to [2], around 60% of the overall video streaming data is mobile. Hence, it is important to adapt video playback methods to the challenges of mobile environments that have a highly volatile throughput due to the variability of the wireless channel (e.g. Wi-Fi or LTE). The variability of LTE throughput over time due to effects like fading, interference, changing number of users etc. has been analyzed in many works see, e.g., [3]–[5]. A commonly used video streaming method is for the server to divide the video into chunks of fixed duration and for the client to download these chunks one after another. The server typically offers these chunks to the client at different bitrates, i.e., different video coding quality. After downloading each chunk, the client estimates the currently available throughput and selects the bitrate of the next chunk based thereon. Examples of such adaptive bitrate streaming methods include Apple's HTTP Live Streaming (HLS) [6] and MPEG-DASH [7].

The client typically maintains a de-jitter buffer where it can store several chunks before playback starts. This pre-buffering can, within certain limits, cope with possible fluctuations of the network conditions. In our work we discuss how to adapt the bitrate of a video stream in order to avoid buffer underflow.

Methods for bitrate adaptation can be roughly divided into two groups: throughput-based and buffer-based. Throughput-based methods like ELASTIC [8] or PANDA [9] often use an estimate of the throughput based on the average performance in the past, whereas buffer-based methods like BOLA [10] consider the current level of the de-jitter buffer at the client (and possibly the throughput in addition) [11].

In this work, we propose a buffer-based adaptive bitrate streaming method that accounts for the variability of throughput by not only using an estimate of the average throughput, but by using an estimate of the throughput distribution. Our method aims at providing the highest possible video bitrate while not exceeding a defined probability of buffer underflow. We use the stochastic network calculus to derive the maximum bitrate for the general case of any throughput distribution. We present analytical and simulation results for the specific example of the throughput having a stationary Gaussian distribution. We have chosen this distribution for the convenience of the numerical evaluation, however, our method also works with models that are commonly used to describe the behavior of wireless channels, such as Rayleigh fading [12].

An analysis of a related problem using similar mathematical methods is presented in [13]. Different from our work, the authors analyze the fill level of a send buffer that stores data for transmission to an LTE base station. Here, the goal is to adapt the transmission resources to prevent the send buffer from overflowing instead of, in our case, adapting the data rate of the source to avoid underflow of the de-jitter buffer at the receiver.

Another related problem is discussed in [14]. Here, the authors analyze a wireless video streaming scenario for which they provide a probabilistic lower bound on the received video quality and use this result for transmission rate adaptation. Different from our work, the authors assume a scenario with strict delay constraints, so that the variability of the wireless channel cannot be compensated by a de-jitter buffer at the receiver.

The main contribution of this work is a model of a de-jitter buffer for video playback that can be used to calculate the probability of buffer underflow. We also show how to calculate the maximum video coding bitrate for which a desired buffer underflow probability is not exceeded. Additionally, we provide simulation results for the concrete example of a Gaussian throughput distribution. Our simulation results show that a system that adjusts the video bitrate according to our method does not experience buffer underflow within the limits of the

978-0-9883045-5-0/18/$31.00 ©2018 ITC
DOI 10.1109/ITC30.2018.10056

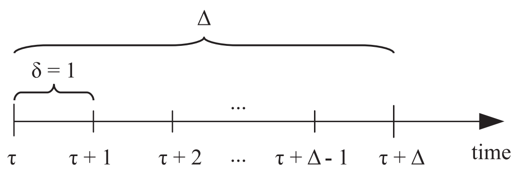


Fig. 1. Time interval $[\tau, \tau + \Delta)$ with granularity $\delta = 1$.

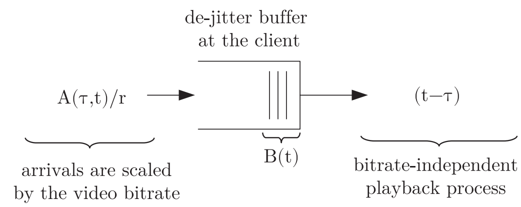


Fig. 2. In the interval $[\tau, t)$ the buffer level $B(t)$ is increased by arrivals $A(\tau, t)$ that have been scaled by the video bitrate r and decreased by the elapsed play back time $(t - \tau)$.

specified probability.

The rest of the paper is organized as follows. In Sec. II we describe the model that we use to derive our method for bitrate selection. We define and calculate the probability of buffer underflow and discuss how the bitrate of a video has to be selected in order to not exceed a desired buffer underflow probability. In Sec. III we present simulation results to demonstrate the performance of the system. The results show that it is possible to achieve a stable bitrate close to the average throughput while maintaining a low probability of buffer underflow. Finally, Sec. IV concludes the paper.

II. Buffer Model

We analyze a model of a video streaming application that maintains a de-jitter buffer in which all downloaded data is stored before being played back. We denote the download process, i.e., the arrivals to the buffer, in the time interval $[\tau, t)$ by $A(\tau, t)$. When the data is played back it is removed from the buffer. For a time interval that starts at time τ and has a duration Δ (e.g., a video chunk), we want to calculate the probability that the buffer level is insufficient, causing an interruption in playback, i.e., buffer underflow. Additionally, we want to calculate the probability that the buffer level stays above some safety margin β after each interval. Our model uses discrete time with time-slot duration δ, see Fig. 1. For notational simplicity, we normalize $\delta = 1$.

In order to determine if a video can be played back without interruption in a time interval $[\tau, \tau + \Delta)$ we have to be able to determine the level of the buffer at any given time $t \in [\tau, \tau+\Delta)$. The reason for this is that we have to make sure that the buffer does not underflow at any point in time. Different from the usual convention, we define our buffer level in units of playback time instead of in units of bits. For a given video bitrate r the scaling is achieved by dividing the number of bits in the buffer by r. Similarly, the arrivals to the buffer and the playback process from the buffer are scaled by division by r. As a consequence, the playback process in the interval $[\tau, t)$ is simply $t - \tau$, regardless of the video bitrate. We use this scaling as it avoids difficulties in the formulation of adaptive bitrate streaming that would otherwise occur, since the buffer may contain parts of different video chunks that are encoded at different bitrates. A general theory of non-constant scaling processes is provided in [15] and following works. Using the scaling by r we can calculate the buffer level at time $t \geq \tau \geq 0$ as

$$B(t) \geq B(\tau) + \frac{A(\tau, t)}{r} - (t - \tau), \tag{1}$$

where $B(t)$ is the buffer level at time t and $B(\tau)$ is the buffer level that was available at time τ. Both $B(t)$ and $B(\tau)$ are measured in units of δ. Eq. (1) is an exact equality if no buffer underflow occurs in $[\tau, t)$, i.e., if the playback is never interrupted. A visual representation of the buffer process can be seen in Fig. 2.

Due to the random nature of wireless channels, the cumulative arrivals to the de-jitter buffer $A(\tau, t)$ in $[\tau, t)$ are random. In each time-slot $[\tau, \tau + 1)$ for $\tau \geq 0$ a random amount of data $X(\tau)$ will be downloaded. We assume that the increments $X(\tau)$ are independent and identically distributed (iid). Each increment has the moment generating function (MGF) $M_X(\theta) = E[e^{\theta X}]$, $\theta \in \mathbb{R}$. We can calculate the cumulative arrivals $A(\tau, t)$ for the interval $[\tau, t)$ as

$$A(\tau, t) = \sum_{i=\tau}^{t-1} X(i).$$

The arrival process is visualized in Fig. 3.

Since the cumulative arrivals $A(\tau, t)$ are random, so is the buffer level $B(t)$. In order to be able to play back a video in an interval $[\tau, t)$ without interruption, the buffer has to stay above a certain threshold b_{min} in every single time step. The value b_{min} can be, e.g., 0 or the size of a single frame. In the following, we want to calculate the probability of a buffer underflow, i.e., the probability of the buffer level falling to or below b_{min}.

Theorem 1 (Buffer Underflow Probability)**.** *Consider a buffer whose fill level is governed by Eq. (1) with iid arrivals that have an MGF $M_X(\theta)$. An upper bound on the probability ε of buffer underflow in the time interval $[\tau, \tau + \Delta]$ is*

$$P\left[\exists t \in [\tau, \tau + \Delta) : B(t) \leq b_{min}\right] \leq e^{-\theta(B(\tau) - b_{min})} := \varepsilon$$

if $\theta > 0$ satisfies the condition

$$M_X\left(\frac{-\theta}{r}\right) e^{\theta} = 1.$$

Proof. A buffer underflow in an interval $[\tau, \tau + \Delta)$ occurs when there exists a time step in which the buffer level is at or below the value b_{min}. Given the interval starts with an initial buffer filling of $B(\tau)$, the probability of this happening follows with Eq. (1) as

$$P[\exists t \in [\tau, \tau + \Delta) : B(t) \leq b_{min}]$$

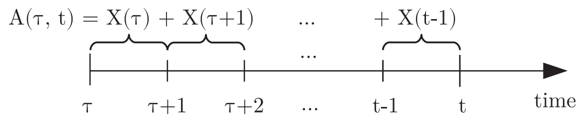


Fig. 3. In each time step a random amount of data $X(i)$ is downloaded. The sum of all downloaded data is $A(\tau, t)$.

$$
\begin{aligned}
&\leq P\left[\min_{t\in[\tau,\tau+\Delta)}\left\{B(\tau)+\frac{A(\tau,t)}{r}-(t-\tau)\right\}\leq b_{min}\right]\\
&= P\left[\max_{t\in[\tau,\tau+\Delta)}\left\{e^{\theta\left((t-\tau)-\sum_{i=\tau}^{t-1}X(i)r^{-1}\right)}\right\}\geq e^{\theta(B(\tau)-b_{min})}\right].
\end{aligned}
\tag{2}
$$

We want to find an upper bound on this probability by using Doob's martingale inequality [16]. The inequality is frequently used in the stochastic network calculus, e.g., [13], [17]–[21]. For this we first need to find a θ for which the process

$$U(t) = e^{\theta\left((t-\tau)-\sum_{i=\tau}^{t-1}X(i)r^{-1}\right)} \tag{3}$$

becomes a martingale, i.e., we need to show that

$$E[U(t+1)|U(t), U(t-1), ..., U(\tau)] = U(t). \tag{4}$$

From Eq. (3) we get $U(t+1) = U(t)e^{\theta\left(1-X(t)r^{-1}\right)}$, which has the conditional expectation

$$
\begin{aligned}
&E[U(t+1)|U(t), U(t-1), ..., U(\tau)]\\
&= U(t)M_X\left(\frac{-\theta}{r}\right)e^{\theta},
\end{aligned}
$$

where we used the independence of the increments $X(i)$. Clearly, $U(t)$ satisfies the condition from Eq. (4) when θ satisfies the constraint from Th. 1.

We can now use a version of Doob's martingale inequality [18, Lem. 2] $P[\max_{t\in[\tau,\tau+\Delta)} U(t) \geq x] \leq E[U(\tau)]x^{-1}$ for non-negative $U(t)$ and $x > 0$, where we shifted the origin of the sequence $U(t)$ from 1 to τ. Application of Doob's inequality to the martingale in Eq. (2) where $x = e^{\theta(B(\tau)-b_{min})}$ and $E[U(\tau)] = 1$ completes the proof. □

The probability ε from Th. 1 can be calculated for general arrivals $A(\tau, t)$ with iid increments $X(i)$ that have the MGF $M_X(\theta)$. We can now calculate the probability from Th. 1 for a specific distribution.

Example 1. In the following, we consider a simple example of a channel that offers a mean throughput μ that varies over time with a variance σ^2. We assume that the throughput of this channel has a Gaussian distribution with $M_X(\theta) = e^{\left(\mu\theta+\frac{\sigma^2\theta^2}{2}\right)}$. Using this MGF we can obtain the value $\theta = 2r(\mu - r)\sigma^{-2}$ for which the constraint of Th. 1 is satisfied. By inserting this value into the definition of ε we obtain the probability of the buffer falling to or below the minimum threshold b_{min} as

$$\varepsilon = e^{-2r(\mu-r)\sigma^{-2}(B(\tau)-b_{min})}. \tag{5}$$

Bitrate Adaptation for b_{min}: In Th. 1, we have shown how to calculate a probability of buffer underflow in a time interval $[\tau, t)$ given iid arrivals $A(\tau, t)$ that are scaled by a bitrate r. We now want to consider the case where we have some desired upper bound on the buffer underflow probability ε, for which we want to calculate the maximum bitrate of the video that does not violate the bound. In the following we call this rate $r_{b_{min}}$, as it is the maximum source bitrate for which the threshold b_{min} is not violated with probability ε. This bitrate can be calculated at the beginning of each interval $[\tau, \tau + \Delta)$ in order to adjust the video stream to the maximum quality for which a buffer underflow is unlikely. Note that the rate $r_{b_{min}}$ can be larger or smaller than the bitrate from the previous interval.

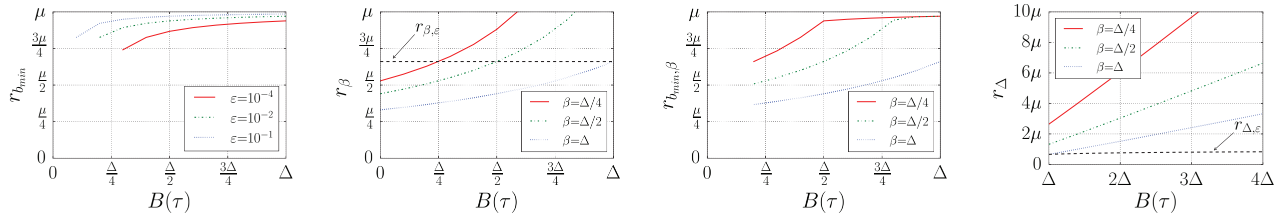
Generally, we can obtain the bitrate $r_{b_{min}}$ for a given arrival MGF by finding a θ that satisfies the condition from Th. 1, inserting this θ into the definition of ε and solving for r. Note that the rate $r_{b_{min}}$ obtained this way is a positive real number. In systems where the rate can only be adjusted in discrete steps, as is the case with, e.g., MPEG-DASH, the bitrate can be chosen as the largest possible bitrate that is lower than $r_{b_{min}}$.

Example 2. For the special case of Gaussian increments that has been introduced in Example 1 we can now obtain the value $r_{b_{min}}$ by solving Eq. (5) for r. To keep the example simple we choose $b_{min} = 0$. We then get

$$r_{b_{min}} = \frac{\mu}{2} + \sqrt{\frac{\mu^2 B(\tau) + 2ln(\varepsilon)\sigma^2}{4B(\tau)}}. \tag{6}$$

Using this example, we now discuss how $r_{b_{min}}$ is affected by the available buffer level $B(\tau)$ and the desired violation probability ε. A graph showing the influence of these parameters on the rate $r_{b_{min}}$ can be seen in Fig. 4a. The x-axis is scaled to the interval length Δ and the y-axis is scaled to the mean download rate μ to emphasize the influence of these parameters on the source bitrate. For small values of $B(\tau)$ the violation probability ε has more influence on the rate $r_{b_{min}}$ than for larger values. For very small values of $B(\tau)$ no rate $r_{b_{min}}$ can be calculated. This is because a certain amount of prebuffering is required for the system to function properly regardless of the rate. Note that the amount of required prebuffering increases when ε becomes more conservative. For any choice of ε the suggested bitrate $r_{b_{min}}$ approaches the mean throughput μ as the buffer level $B(\tau)$ increases. The rate never exceeds μ as that would violate the condition $\theta > 0$.

Safety Margin: As we have seen in Eq. (6) and Fig. 4a, some prebuffering is required to achieve a desired violation probability ε - e.g. it should be about $\frac{\Delta}{4}$ for $\Delta = 10$ and $\varepsilon = 10^{-2}$. Besides, some additional safety margin might be desirable in order to avoid application failure in case the average download rate changes rapidly. This safety margin can be reached by ensuring that a certain target buffer level $\beta > 0$ is available at the end of a time interval $[\tau, \tau + \Delta)$. We can



(a) $r_{b_{min}}$ is the maximum rate allowed so that the buffer does not underflow at any time step.

(b) r_β is the maximum rate allowed so that a safety margin β is achieved at time $\tau + \Delta$. $r_{\beta,\varepsilon}$ is the minimal rate with probability $1-\varepsilon$.

(c) $r_{b_{min},\beta}$ is the minimum of the two rates $r_{b_{min}}$ and r_β.

(d) $r_\Delta$is the maximum rate allowed so that the a safety margin β is achieved at time $\tau + n\Delta$. $r_{\Delta,\varepsilon}$ is the minimal rate with probability $1-\varepsilon$.

Fig. 4. Graphs of the rates $r_{b_{min}}$, r_β, $r_{b_{min},\beta}$ and r_Δ for parameters $X(i) \sim \mathcal{N}(4,2)$, $\Delta = 10$, $b_{min} = 0, \varepsilon = 10^{-2}$.

generalize this for longer intervals of length $n\Delta$, $n \geq 1$ by guaranteeing this safety margin at the end of a longer time interval $[\tau, \tau + n\Delta)$, i.e., $B(\tau + n\Delta) \geq \beta$. Note that since $B(\tau)$ is a scaled buffer in units of δ, the safety margin β is also in units of δ. In the following we want to calculate the probability of violating the safety margin β at the end of an interval.

Theorem 2 (Probability of Violating the Safety Margin β)**.** *Consider a buffer whose fill level is governed by Eq. (1) with iid arrivals that have an MGF $M_X(\theta)$. An upper bound on the probability ε of violating the safety margin β at the end of the interval $[\tau, \tau + n\Delta)$ is*

$$\begin{aligned}&P[B(\tau + n\Delta) \leq \beta]\\ &\leq \min_{\theta \geq 0} \left\{ e^{\theta r(\beta + n\Delta - B(\tau))} M_X(-\theta)^{n\Delta} \right\} := \varepsilon.\end{aligned}$$

Proof. In order to calculate the probability of violating the safety margin β at the end of the time interval $[\tau, \tau + n\Delta)$ it is enough to only observe the buffer level at the end of the interval, i.e., at time $t = \tau + n\Delta$. The probability of violating the safety margin β follows with Eq. (1) as

$$\begin{aligned}&P[B(\tau + n\Delta) \leq \beta]\\ &\leq P[A(\tau, \tau + n\Delta) \leq r(\beta + n\Delta - B(\tau))].\end{aligned}$$

By applying Chernoff's bound $P[X \leq x] \leq e^{\theta x} E\left[e^{-\theta X}\right]$ for $\theta \geq 0$ and using the independence of the increments $X(i)$ we directly obtain the probability in Th. 2. □

The probability in Th. 2 can be calculated for arbitrary arrivals $A(\tau, \tau + n\Delta)$ with iid increments $X(i)$ which have the MGF $M_X(\theta)$. If we select a specific distribution of the increments, we can calculate the value θ for which this probability becomes minimal.

Example 3. For Gaussian increments, which we defined in Example 1, the probability from Th. 2 is minimal for

$$\theta = \frac{n\Delta\mu - r(\beta + n\Delta - B(\tau))}{n\Delta\sigma^2}$$

By inserting this value of θ into the definition of ε we get

$$\varepsilon = \exp\left(\frac{-(n\Delta\mu - r(\beta + n\Delta - B(\tau)))^2}{2n\Delta\sigma^2}\right)$$

as the probability for violating the safety margin β at the end of the interval $[\tau, \tau + n\Delta)$.

Bitrate Adaptation for β in $[\tau, \tau + \Delta)$: After obtaining the probability for violating the safety margin β we can calculate r_β, which is the maximal video bitrate that ensures that the buffer level is at least β at the end of the interval $[\tau, \tau + \Delta)$ with a violation probability ε.

Example 4. Using the probability from Example 3 for $n = 1$ and $B(\tau) < \Delta$ we can calculate

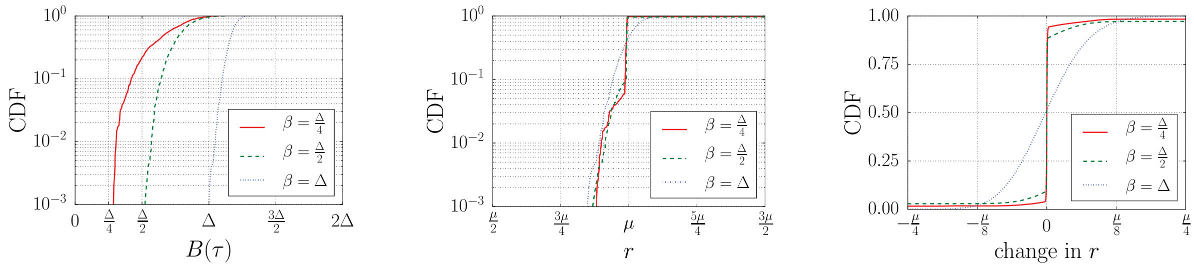
$$r_\beta = \frac{\Delta\mu - \sqrt{-2\Delta \ln(\varepsilon)\sigma^2}}{\beta + \Delta - B(\tau)}. \tag{7}$$

We can also calculate a minimum source bitrate $r_{\beta,\varepsilon}$ from Eq. (7) that applies if $B(\tau)$ equals β. Since $B(\tau) \geq \beta$ with probability $1 - \varepsilon$, the rate r_β falls below $r_{\beta,\varepsilon}$ at most with probability ε. In Fig. 4b, we present a graph of the rate r_β for different values of β. It can be seen that r_β grows with increasing $B(\tau)$ without being constrained by μ as was the case for $r_{b_{min}}$. This is because, different from $r_{b_{min}}$, a specific level of the buffer has to only be guaranteed at the end of the large interval $[\tau, \tau + \Delta)$ and not at every single time step, therefore as $B(\tau)$ increases, the probability of not having a buffer level β at time $\tau + \Delta$ approaches 0. The black dashed line indicates the minimum rate $r_{\beta,\varepsilon}$. Note that $r_{\beta,\varepsilon}$ is independent of β, meaning the minimum buffer level has no effect on the minimum source bitrate. The parameter β only influences how much the source bitrate changes with increasing $B(\tau)$, the changes being higher for less conservative values of β.

In the period $[\tau, \tau + \Delta)$ both constraints b_{min} and β have to be satisfied. For this, we have to select the minimum of the two bitrates:

$$r_{b_{min},\beta} = \min\{r_{b_{min}}, r_\beta\}. \tag{8}$$

In Fig. 4c, we present a graph of the source bitrate $r_{b_{min},\beta}$ for different values of β and $\varepsilon = 10^{-2}$. It can be seen that for smaller values of $B(\tau)$ the rate r_β is the minimum of the two rates, whereas for larger values the minimum is generally $r_{b_{min}}$. The switching point between the two rates depends on the choice of β: for less conservative values the switch happens earlier than for the more conservative ones (if at all) since the



(a) Stationary distribution of $B(\tau)$ (b) Stationary distribution of r (c) Variability of r between intervals

Fig. 5. Effect of different values of the safety margin β on the buffer level $B(\tau)$ and the source bitrate r for $X(i) \sim \mathcal{N}(4, 2)$, $\Delta = 50$, $b_{min} = 0, \varepsilon = 10^{-2}$ and $\beta \in \left\{\frac{\Delta}{4}, \frac{\Delta}{2}, \Delta\right\}$.

source bitrate is adjusted towards the average throughput more quickly.

Bitrate Adaptation for β in $[\tau, \tau + n\Delta)$: In the case that the buffered data is sufficient for the upcoming interval $[\tau, \tau + \Delta)$, meaning $B(\tau) \geq \Delta$, we no longer have to adjust the source bitrate to ensure that the buffer does not fall below the threshold b_{min} at some time t in $[\tau, \tau + \Delta)$. This has the advantage that we can select a source bitrate that is no longer bounded by the mean download rate μ as has been the case with $r_{b_{min}}$ in Fig. 4a. Instead, we can now choose a new source bitrate r_Δ in a way that ensures that the safety margin β is achieved after a larger interval $[\tau, \tau + n\Delta)$ with probability $1 - \varepsilon$, where $n = \frac{B(\tau)}{\Delta}$, $n \in \mathbb{R}$.

Example 5. Similar to Example 3, for Gaussian increments we can find a θ for which ε from Th. 2 is minimal and solve for r_Δ in order to obtain the maximum bitrate for which $B(\tau+n\Delta) \geq \beta$ with violation probability ε:

$$r_\Delta = \frac{n\Delta\mu - \sqrt{-2n\Delta\ln(\varepsilon)\sigma^2}}{\beta + n\Delta - B(\tau)}. \tag{9}$$

Again, we can calculate a minimum bitrate $r_{\Delta,\varepsilon}$ from Eq. (9) that applies if $B(\tau)$ equals β. Since $B(\tau) \geq \beta$ with probability $1 - \varepsilon$, the rate r_Δ falls below $r_{\Delta,\varepsilon}$ at most with probability ε. The resulting graph of r_Δ for different values of β and $\varepsilon = 10^{-2}$ is shown in Fig. 4d. The bitrate grows linearly with the available buffer $B(\tau)$. The slope of the rate is smaller for more conservative choices of β. Because the buffer level $B(\tau)$ is very large, the resulting source bitrate is almost always larger that μ. The black dashed line indicates the minimum source bitrate $r_{\Delta,\varepsilon}$. This rate is independent from β, it does however grow very slightly with $B(\tau)$.

Finally, for all values of $B(\tau)$ the source bitrate selection can be written as

$$r = \begin{cases} r_{b_{min},\beta} & \text{, for } B(\tau) < \Delta \\ r_\Delta & \text{, for } B(\tau) \geq \Delta. \end{cases} \tag{10}$$

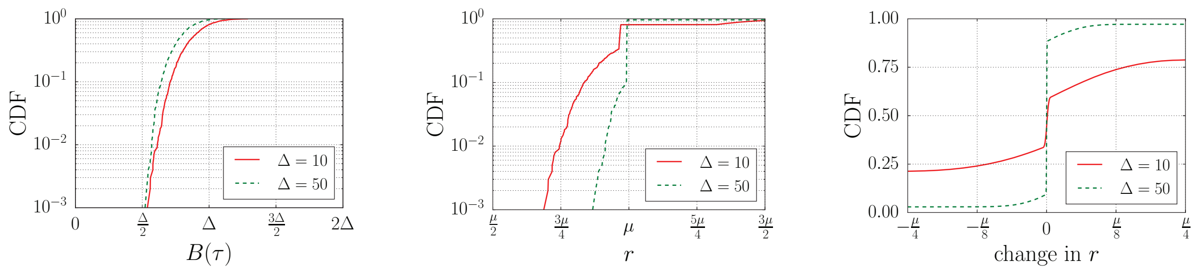
III. Simulation Results

In this section, we present and discuss simulation results in order to illustrate the performance of the system that adapts its bitrate according to the method described in Eq. (10). For this, we have written a discrete-time simulator. In each time step δ the buffer level is increased by a Gaussian increment $X(i) \sim \mathcal{N}(\mu, \sigma^2)$ that is scaled by the current source bitrate r, and decreased by δ. Each simulation run is configured to consist of 10^3 intervals Δ where $\Delta = 50\delta$.

In all performed simulation runs both the source bitrate r and the buffer level $B(\tau)$ over time converged to a stationary distribution within the first few intervals Δ. The shape of these distributions depends on the parameters $\mu, \sigma^2, b_{min}, \beta, \Delta$, and ε. In the following we are going to discuss how some of these parameters affect the resulting distribution by presenting the corresponding simulation results. For each set of parameters we perform 10^3 simulation runs and show the distribution of the results.

First we discuss the effect of β on the resulting distributions of the buffer level $B(\tau)$ and the source bitrate r. In Fig. 5a, it can be seen that with increasing target level β the distribution of the buffer level also becomes larger and less variable. Note that for the selected probability $\varepsilon = 10^{-2}$ the buffer level $B(\tau)$ satisfies both specified constraints β and b_{min}.

In Fig. 5b, we can see that for small values of β the source bitrate stays stable in about 90% of the cases and there are not many differences in the distributions based on β. For $\beta = \Delta$ the distribution becomes more variable, however the median remains the same. The variability is small for buffer levels $B(\tau) < \Delta$ because the source bitrate is selected according to Eq. (8) as shown in Fig. 4c, making it close to the mean download rate μ and changing only very slightly with changes in $B(\tau)$. For $B(\tau) \geq \Delta$ the source bitrate is selected as shown in Fig. 4d, meaning it increases and decreases linearly with any change in $B(\tau)$, resulting in higher variability. We further investigate the effect of the parameter β on the variability of the source bitrate in Fig. 5c by plotting the distribution of the changes between neighboring intervals. It can be seen that for smaller values of β only slight changes of the source bitrate should be expected when going from one interval to the next, with large changes being very rare. For $\beta = \Delta$ the source bitrate changes with every new interval, however it stays the same on average. This occurs because of the linear behavior of r_Δ that can be seen in Fig. 4d. A more conservative rate



(a) Stationary distribution of $B(\tau)$

(b) Stationary distribution of r

(c) Variability of r between intervals

Fig. 6. Effect of different values of the interval length Δ on the buffer $B(\tau)$ and the source bitrate r for $X(i) \sim \mathcal{N}(4, 2)$, $b_{min} = 0, \varepsilon = 10^{-2}$, $\beta = \frac{\Delta}{2}$ and $\Delta \in \{10, 50\}$.

selection method for large safety margins $\beta \geq \Delta$ may remedy these rate fluctuations.

The interval length Δ also has an effect on the distributions of $B(\tau)$ and r. The comparison of the buffers in relation to Δ in Fig. 6a tells us, that for higher values of Δ the buffer is slightly smaller, which is due to the median source bitrate being higher, as can be seen in Fig. 6b. Here we can also see that for the higher value of Δ the source bitrate r remains stable with a higher probability, whereas for the smaller Δ the source bitrate is more variable. The influence of Δ on rate changes between intervals can be seen in Fig. 6c: For the larger Δ the probability of a rate change is very small, whereas for a smaller Δ this probability increases. Overall, it can be said that choosing a larger Δ yields a more stable distribution of bitrates as the variability of the individual time steps becomes less important over longer intervals. Note that increasing Δ also means that the bitrate can be adapted less frequently.

IV. Conclusion

In this paper, we have presented and analyzed a buffer-based bitrate adaptation technique for variable bitrate video streaming. We have derived a buffer underflow probability for the general case of the arrivals having any iid distribution and shown how this probability can be used for bitrate selection using an example of a specific distribution. Additionally, we have calculated a minimum possible bitrate that is only violated with a small probability when the rate is adjusted according to our method. Using a series of simulations we have analyzed the impact of different system parameters on the selected bitrate. In all performed simulation runs, regardless of the initial buffer level, both the bitrate and the buffer level converged to stationary distributions after only few intervals. For all selected parameters the simulated system did not violate the specified buffer underflow probability. We have also observed that the bitrate distribution becomes more stable and that the minimum bitrate becomes larger when the interval length Δ increases, i.e., when adaptation occurs less frequently. We have shown that our system requires some minimum buffer level which depends on the desired maximal probability of buffer underflow, i.e., video stalling. Due to the way our method adjusts the bitrate, the bitrate distribution becomes more variable with an increase of the desired safety margin. This might be different for more conservative bitrate selection methods.

References

[1] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021,” 2016.
[2] Ooyala, “Global Video Index Q4 2017.”
[3] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, “A close examination of performance and power characteristics of 4G LTE networks,” in *Proc. of ACM MobiSys*, 2012.
[4] J. Cainey, B. Gill, S. Johnston, J. Robinson, and S. Westwood, “Modelling download throughput of LTE networks,” in *Proc. of IEEE LCN Workshops*, 2014.
[5] M. Akselrod, N. Becker, M. Fidler, and R. Lübben, “4G LTE on the Road-What Impacts Download Speeds Most?” in *Proc. of the IEEE VTC-Fall*, 2017.
[6] “Apples HTTP Live Streaming,” https://developer.apple.com/streaming/, accessed: 2018-04-22.
[7] T. Stockhammer, “Dynamic adaptive streaming over HTTP- standards and design principles,” in *Proc. of ACM MMSys*, 2011.
[8] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, “ELASTIC: a Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH),” in *Packet Video Workshop (PV)*, 2013.
[9] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, “Probe and adapt: Rate adaptation for HTTP video streaming at scale,” *IEEE JSAC*, vol. 32, no. 4, pp. 719–733, Apr. 2014.
[10] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, “BOLA: Near-optimal bitrate adaptation for online videos,” in *Proc. of IEEE INFOCOM*, 2016.
[11] T. C. Thang, H. T. Le, A. T. Pham, and Y. M. Ro, “An evaluation of bitrate adaptation methods for HTTP live streaming,” *IEEE JSAC*, vol. 32, no. 4, pp. 693–705, Apr. 2014.
[12] H. Al-Zubaidy, J. Liebeherr, and A. Burchard, “Network-Layer Performance Analysis of Multihop Fading Channels,” *IEEE Trans. on Networking*, vol. 24, no. 1, pp. 204–217, Feb. 2016.
[13] A. Rizk and M. Fidler, “Queue-aware uplink scheduling with stochastic guarantees,” *Computer Communications*, vol. 84, pp. 63–72, Jun. 2016.
[14] H. Al-Zubaidy, V. Fodor, G. Dán, and M. Flierl, “Reliable Video Streaming With Strict Playout Deadline in Multihop Wireless Networks,” *IEEE Trans. on Multimedia*, vol. 19, no. 10, pp. 2238–2251, Aug. 2017.
[15] M. Fidler and J. B. Schmitt, “On the way to a distributed systems calculus: A network calculus with data scaling,” in *Proc. of ACM SIGMETRICS*, Jun. 2006, pp. 287–298.
[16] J. L. Doob, *Stochastic processes*. Wiley, 1953.
[17] J. F. C. Kingman, “A martingale inequality in the theory of queues,” *Math. Proc. Cambridge*, vol. 60, no. 2, pp. 359–361, Apr. 1964.
[18] Y. Jiang, “A note on applying stochastic network calculus,” 2010.
[19] ——, “Network calculus and queueing theory: Two sides of one coin,” in *VALUETOOLS*, 2009.
[20] M. Fidler, B. Walker, and Y. Jiang, “Non-asymptotic delay bounds for multi-server systems with synchronization constraints,” *IEEE Trans. on Parallel and Distributed Systems*, Jan. 2018.
[21] F. Ciucu and J. Schmitt, “Perspectives on network calculus: no free lunch, but still good value,” in *Proc. of ACM SIGCOMM*, Aug. 2012, pp. 311–322.