



HAL
open science

Appendices on QoE model and DASH manifest in "Quality of Experience-based Routing of Video Traffic for Overlay and ISP Networks"

Giacomo Calvigioni, Ramon Aparicio-Pardo, Lucile Sassatelli, Jérémie
Leguay, Paolo Medagliani, Stefano Paris

► To cite this version:

Giacomo Calvigioni, Ramon Aparicio-Pardo, Lucile Sassatelli, Jérémie Leguay, Paolo Medagliani, et al.. Appendices on QoE model and DASH manifest in "Quality of Experience-based Routing of Video Traffic for Overlay and ISP Networks". [0] Université Côte d'Azur. 2018. hal-01672042

HAL Id: hal-01672042

<https://hal.science/hal-01672042v1>

Submitted on 22 Dec 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Appendices on QoE model and DASH manifest in "Quality of Experience-based Routing of Video Traffic for Overlay and ISP Networks"

Giacomo Calvigioni, Ramon Aparicio-Pardo, Lucile Sassatelli Universite Cote d'Azur, CNRS, I3S

Email: {calvigioni,raparicio,sassatelli}@i3s.unice.fr

Jeremie Leguay, Paolo Medagliani, Stefano Paris France Research Center, Huawei Technologies Co. Ltd

Email: {name.surname}@huawei.com

Abstract

This report provides the details about (i) the QoE-QoS model and (ii) the DASH representations used in the simulations performed in [2]. The report details the calculations required to find a concave Quality of Experience (QoE) curve from Quality of Service (QoS) metrics by using the reference model of ITU-T Rec. P.1202.2 [3], [6], the calibration and linearization of such QoE curve; and, the Dynamic Adaptive Streaming over HTTP (DASH) representations used in the paper simulations.

I. QOE-QOS MODEL

We used the reference model of ITU-T Rec. P.1202.2 [3], [6] that takes into account QoS metrics within a log-logistic model. Some latent variables, $\{d\}$, have been introduced to denote the distortion due to each type of impairment, and the final quality score comes from a combination of all the distortions. The latent variables often are not measurable straightforwardly. However, given a set of videos which suffer from only one type of (i.e., uni-type) impairment, is possible to learn the quality attribute as only one latent variable is activated (i.e., taking a nonzero value). Thus, is possible to learn the quality model in two steps. In the first step, the quality attribute function is determined for each uni-type impairment. In the second step, the total function on video samples with hybrid impairments is optimized, under the constraints that the partial form of the total function should match the attribute functions determined in the first step.

Without loss of generality, we consider that there are three types of impairments, denoted by c , s , and f respectively. The above process can be formulated as: in the first step, an investigation of the quality function $f_c(d_c|d_s = d_f = 0)$ for c ,

$f_s(d_s|d_c = d_f = 0)$ for s , and $f_f(d_f|d_c = d_s = 0)$ for f ; in the second step, we explore the total function $f(d_c, d_s, d_f)$ under the following constraints

$$\begin{cases} f_c(d_c|d_s = d_f = 0) = f_c \\ f_s(d_s|d_c = d_f = 0) = f_s \\ f_f(d_f|d_s = d_c = 0) = f_f \end{cases} \quad (1)$$

Skipping the details on how to extract the features for each uni-type impairment and using the formulations in [5] with coefficients taken from [3], [6], we can determine a function connecting the network parameters to the subjective parameters related to user experience. The analyzed framework presents a technique to generate a metric space in which the video distortions, visual problems perceived by the user seeing the video, are addable. In particular, the studied model presents three types of impairments that a video can have, each one characterized by a key-factor and a co-variate:

- 1) **Compression:** formation of uniform artifacts in video image that decrease the general clearness of the video frame. It is known as texture mask effect:
 - Key factor: $51 - QP$ where QP is the video quantization parameter
 - Co-variate: Content Unpredictability (CU)
- 2) **Freezing:** interruption of the video reproduction due to buffer exhaustion or insufficient bandwidth
 - Key factor: Freezing Duration (FD)
 - Co-variate: Content Unpredictability (CU)
- 3) **Slicing:** generation of so called mosaic artifacts due to reparation of damaged frames using neighbor pixels' information
 - Key factor: Error Rate (ER)
 - Co-variate: Content Unpredictability (CU)

These three types of distortion can be used to calculate the quality of experience perceived by the user. The corresponding framework can be formulated as follows:

$$f = \frac{1}{1 + \left(a'_c z'_c x'_{c0} + a'_s z'_s x'_{s0} + a'_f z'_f x'_{f0} \right)^\beta} \quad (2)$$

where $a_i, i = \{c, s, f\}$ are the key factors and $z_i, i = \{c, s, f\}$ are the co-variates. Given this framework, we decided to take in consideration just two of the impairments presented, compression and freezing, that are the two main factors affecting mostly the user QoE, and find the relationship between the QoS metrics and these impairments.

Compression: the video compression is calculated using the variable $51 - QP$ where QP is the quantization parameter of the actual video chunk. The higher the quantization parameter, the lower the video quality, being the picture more compressed

and with more visual artifacts. This metric is strictly related to the bit-rate change ratio.

QP is directly related to the user available bandwidth or, more specifically, to the download time of the previous chunk. In the framework, the compression factor x_c , is calculated as $51 - QP$. Thus, knowing the characteristics of the video, resolution and frame rate, it is possible to calculate the required amount of bandwidth related to a specific compression level. This relationship directly connects the quantization parameter to the user available bandwidth. Being QP a discrete value in range between 1 and 51, and knowing the range of requested bandwidth for a given video resolution and frame rate, it is possible to map these two ranges with a geometrical transformation and calculate the compression level along a video with the following formula:

$$x_c^n = \frac{\sum_{k=1}^{k(n)} 51 - QP(br(q_n^k))}{k(n)} \quad \forall n \leq N \quad (3)$$

where n is the n -th user, N the number of requests (users) in the network, k is the video chunk index, $k(n)$ is the number of chunks for the n -th request, q_n^k is the selected quality level for chunk k and request n , $(br(q_n^k))$ is the bit rate of the selected quality level, $QP(br(q_n^k))$ is the quantization parameter bit rate of the selected chunk (strictly related to the bandwidth assigned to each request), 51 is the maximum compression according in AVC (H.264) and 1 is the minimum compression.

Content Unpredictability: the content unpredictability (CU) is the covariate of the compression factor and represents the spatial-temporal complexity of a video. It is calculated on each Macro Block of each frame in the video as the variance of the residuals in the luminance channel. The higher the value, the more complex the video. Human eye is prone to tolerate visual distortion in complex scenes, but the user will notice it only inside a simple scene. The CU of each Macro Block (MB) is calculated using the following formula:

$$CU = \frac{1}{K_{MB}} \sum_{k=1}^{K_{MB}} \left(I_{r,k} - \sum_{k=1}^{K_{MB}} I_{r,k} \right)^2$$

where r is the index of the MB K_{MB} is the number of MBs in a frame and $I_{r,k}$ is the residual of luminance channel of the k -th residual pixel in the r -th MB.

Freezing: the freezing duration considered in this framework is defined as the sum of all freezing events, measured in seconds, perceived by the user, normalized over the total duration of the video. As the metric presented in [6] does not take in account the buffer that the user has accumulated during the download of previous chunks, we extended it using the recursive formula presented in [5], in order to calculate the residual buffer in seconds at chunk k :

$$B_{k+1} = \left(\left(B_k - \frac{S_k(x_c^n)}{\sum_{p \in P} y_p^n} \right)_+ + L \right)_+ \quad \forall k \leq k(n) \quad (4)$$

where B is the buffer in seconds after the download of the previous chunk, $S_k(x_c^n)$ is size of the actual chunk related to the selected compression level x_c^n , $\sum_{p \in P} y_p^n$ is the allocated bandwidth over all paths from source of the video to the client, and L is the length of the chunk (in seconds).

Using the same notation, we can compute the total freezing duration of video by summing all the freezing events that the user will experience given a fixed amount of allocated bandwidth. A freezing event occurs when the time required to download a chunk is larger than the sum of the cumulated buffer plus the duration of a chunk; in this case, the decoder goes in freezing mode and the actual chunk is replaced with the last correctly downloaded chunk. The number of skipped chunks for each request is then normalized to the number of video chunks.

$$x_f^n = \frac{\sum_{k=1}^{k(n)} \left[\left(\frac{S_k(x_c^n)}{\sum_{p \in P} y_p^n} - B_k - L \right)_+ / L \right]^{c_4}}{k(n)}, \forall n \leq N \quad (5)$$

where c_4 is a parameter related to the temporal pooling strategy set to 0.9 after grid optimization.

Motion Homogeneity (MH) Motion homogeneity (MH) is the covariate of freezing factor and quantifies the amount of movements in the scene affected by the freezing effect. If the freezing happens in a scene with absence of movement, it will not be noticed by the user, giving a small impact on the overall perceived QoE. MH is defined as the maximum between Isotropic Homogeneity (IH) and Radial Homogeneity (RH) for each frame; the first one is greater when the camera is tracking some object or panning or tilting and the second is greater when the scene is zooming or dolling. MH is calculated for all the frames involved by the freezing event with the following formulas:

$$MH_p = \sum_{\tau=1}^P \max\{IH_\tau, RH_\tau\}$$

where

$$IH_\tau = \frac{1}{HW} \sqrt{\left(\sum_{r \in \tau} \sum_{l \in \tau} MV_{h,l,r} A_{l,r} \right)^2 + \left(\sum_{r \in \tau} \sum_{l \in \tau} MV_{v,l,r} A_{l,r} \right)^2}$$

and

$$RH_\tau = \frac{1}{HW} \sqrt{\left| \sum_{r \in \tau_L} \sum_{l \in r} MV_{h,l,r} A_{l,r} - \sum_{r \in \tau_R} \sum_{l \in r} MV_{h,l,r} A_{l,r} \right|^2 + \left| \sum_{r \in \tau_T} \sum_{l \in r} MV_{v,l,r} A_{l,r} - \sum_{r \in \tau_B} \sum_{l \in r} MV_{v,l,r} A_{l,r} \right|^2}$$

τ is the index of pauses (i.e., the freezing events), P is the number of pauses, H is the picture height (in number of pixels), W is the picture width (in number of pixels), l is the MB partition index, r is the MB index, $MV_{h,l,r}$ is the horizontal motion vector of partition l in MB r , $A_{l,r}$ is the area of partition l in MB r (in number of pixels), $MV_{v,l,r}$ is the vertical motion vector of partition l in MB r , τ_L is the frame left half, τ_R is the frame right half, τ_T is the frame top half, and τ_B is the frame bottom half.

Now that we have defined the average compression and the total freezing duration of the video, we can write the equation to calculate the QoE perceived by the user, given the allocated bandwidth, as follows:

$$QoE_n = \frac{1}{1 + \alpha \left(a'_{c_n} z'_{c_1} b'_{c_0} x_{c_n} + a'_{f_n} z'_{f_1} b'_{f_0} x_{f_n} \right)^\beta}, \forall n \leq N \quad (6)$$

where x_c and x_f are respectively the average compression of the video and the total freezing duration (i.e., the main factors), z_c is the content unpredictability, z_f is the motion homogeneity (i.e., the two co-factors), a , b , α , and β are the parameters used in this function to maintain the logistic shape of the curve. These parameters are calculated using a set of pre-distorted video as presented in [3], [6].

The following table contains the coefficients values for different resolutions from the ITU-T recommendation [3], [6] trained with machine-learning techniques:

Resolution	a_c	$\ln(a_f)$	b_{c_0}	b_{c_1}	b_{f_0}	b_{f_1}	$\ln(\alpha)$	β	c_4
360p	1	-4.45	-2.51	-0.262	1.18	0.137	4.82	0.734	0.9
720p	1	-2.03	-1.35	-0.144	0.985	0.087	4.6	1.27	0.9
1080p	1	-1.31	-1.21	-0.147	0.908	0.118	3.32	1.15	0.9

II. MODEL CALIBRATION AND LINEARIZATION

In order to evaluate the QoE curve, we consider Big Buck Bunny (BBB), a reference video used for calibration. Several video bit rates are used: {350, 470, 630, 845, 1130, 1520, 2040, 2750} Kb/s, 299 chunks, with a chunk length of 2 seconds and a highly variable chunk size. The buffer size varies in the following interval: $T = \{2, 5, 10, 30, 60\}$ s.

As shown in Figure 1, the shape of the QoE curve is highly non-linear. In order to introduce it in an optimization model, we must linearize it first. As Equation 6 has a log-logistic shape, it is possible to linearize it using a piece-wise linear function, as presented in [1].

Considering the following maximization problem

$$f(x) = \max_{i=1, \dots, m} (a_i^T x + b_i)$$

This problem can be transformed to

$$\min t$$

$$\text{s.t.}$$

$$a_i^T x + b_i \leq t \quad i = 1, \dots, m$$

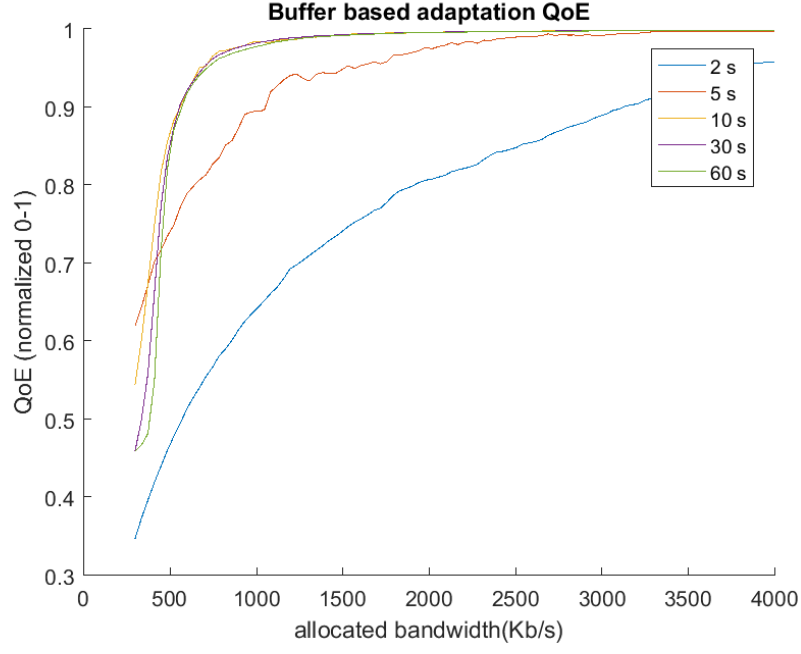


Figure 1. QoE extracted for Big Buck Bunny video.

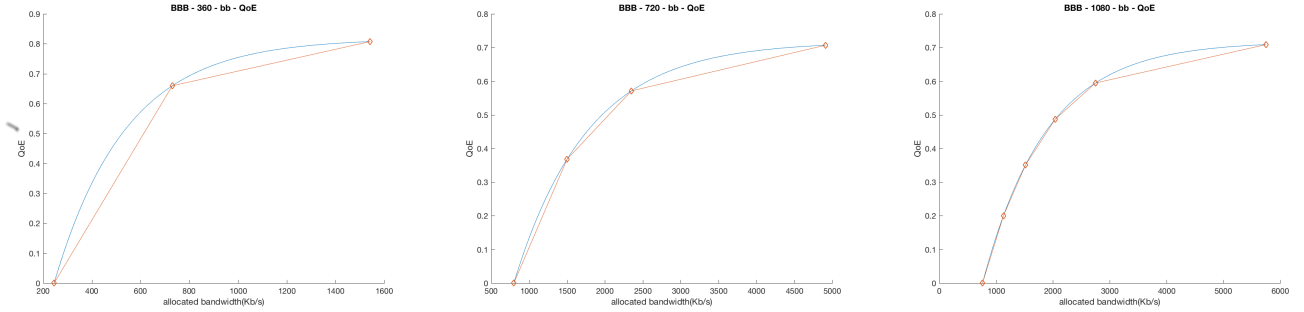


Figure 2. Linearized QoE of three different resolutions of the BBB video, namely 360p, 720p, and 1080p.

where each piece of the function generates a set of constraints using as parameters the parameters of the corresponding linear function. The linear problem then becomes

$$\max_{q_n, y_p^n} \sum_{n \in N} q_n \quad (7)$$

s.t.

$$\sum_{p \in P_e} y_p^n \leq b_e \quad \forall e \in \varepsilon \quad (8)$$

$$a_k \sum_{p \in P_n} y_p^n + b_k \geq q_n \quad \forall k \in K, n \in N$$

We applied the linearization presented in Equation 7 to three different resolutions of the BBB video (namely 360p, 720p, and 1080p). The results are presented in Figure 2.

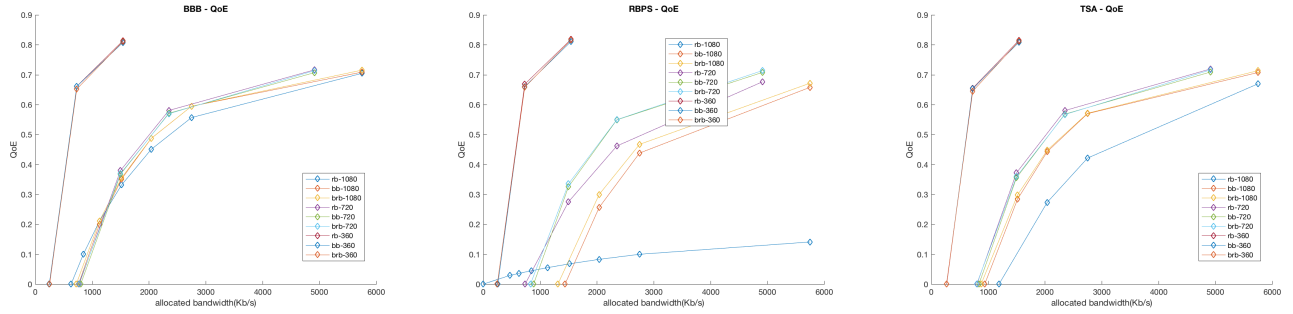


Figure 3. Linearized QoE of three different resolutions (360p, 720p, and 1080p) of the BBB video (a), RBPS video (b), and TSA video (c). Three buffer adaption policy are considered: buffer based (bb), rate based (rb), and buffer and rate based (brb).

The linearization presented above, denoted as bb, has been extended for other types of DASH adaptation policy, i.e., rate based (rb) and buffer and rate based (brb), an hybrid combination of buffer and rate based. We have considered three types of videos, namely Big Buck Bunny (BBB), Red Bull Play Street (RBPS), and The Swiss Account (TSA). The aggregated results of the linearization are shown in Figure 3. The same three video resolutions are considered.

III. DASH REPRESENTATIONS USED IN THE SIMULATIONS

We use the Adaptive Multimedia Streaming Simulator Framework (AMust)¹ [4] in ns-3 which implements an HTTP client and server for LibDASH², one of the reference software of ISO/IEC MPEG-DASH standard.

We considered three representative open movies³ commonly used for testing video codecs and streaming protocols and recommended in the measurement guidelines of the DASH Industry Forum: Big Buck Bunny (BBB), high motion computer animated movie, Swiss Account (TSA), which is a sport documentary with regular motion scenes and Red Bull Play Street (RBPS), which is a sport show with high motion scenes.

While the Named Data Networking (NDN) module of AMust supports the streaming of real video segments, the IP module does not. In both cases, they can stream fake video segments from a given set of representations at targeted bitrates. We have then extended this code to consider the chunk size variability observed in reality for the three movies. As the distribution of chunk size showed is heavy-tailed, we have approximated for each representation the chunk size distribution with a log normal distribution and estimated its parameters: average and ratio σ/μ .

One can find below all the parameters used for the generation of chunks in the ns-3 simulations for BBB, TSA and RBPS.

¹<https://github.com/ChristianKreuzberger/AMuSt-Simulator/>

²<https://github.com/bitmovin/libdash>

³<http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/>

A. DASH representations for Big Buck Bunny (BBB)

```

segmentDuration=2s
numberOfSegments=298
AvgSigma/mu=0.041367296119656
reprId,QualityIndex,screenWidth,screenHeight,bitrate,sigma/mu,avgchunksiz
1,0.25,480,360,80,0.0470065072288359,124.936053691275
2,0.50,480,360,165,0.0387431033244563,252.830067114094
3,0.75,480,360,375,0.0354448182241559,583.462093959732
4,1.00,480,360,730,0.0369201955966515,1099.11283221477
5,0.17,1280,720,300,0.0459299846170403,612.372483221476
6,0.33,1280,720,420,0.0419317129876795,814.156832214765
7,0.50,1280,720,585,0.0388625623822102,1095.73594630872
8,0.67,1280,720,920,0.0356634458851894,1674.82655033557
9,1.83,1280,720,1500,0.0340392878281805,2672.29157046980
10,1.00,1280,720,2350,0.0345314429329601,4068.04303355705
11,0.125,1920,1080,350,0.0585012656841168,970.322362416108
12,0.250,1920,1080,470,0.0538216576563194,1187.94389261745
13,0.375,1920,1080,630,0.0493680511156428,1479.61259060403
14,0.500,1920,1080,875,0.0459064981730999,1880.63154362416
15,0.625,1920,1080,1130,0.0430111935057322,2427.11495302014
16,0.750,1920,1080,1520,0.0405682573935096,3197.75307382551
17,0.875,1920,1080,2040,0.0390528773049721,4252.67978523490
18,1.000,1920,1080,2750,0.0384114751124629,5654.20193288591

```

B. DASH representations for Red Bull Play Street (RBPS)

```

segmentDuration=2s
numberOfSegments=299
AvgSigma/mu=0.025216492192589
reprId,QualityIndex,screenWidth,screenHeight,bitrate,sigma/mu,avgchunksiz
1,0.25,480,360,80,0.0332467635352976,155.286367892977
2,0.50,480,360,165,0.0239040736301705,299.272080267559
3,0.75,480,360,375,0.0186760194089483,688.493698996655
4,1.00,480,360,730,0.0170586507642984,1338.57233444816
5,0.17,1280,720,300,0.0329088397520866,843.967973244147
6,0.33,1280,720,420,0.0292248481571937,1091.36628762542
7,0.50,1280,720,585,0.0257326345323223,1439.32521739130
8,0.67,1280,720,920,0.0214581894588418,2171.80436120401
9,0.83,1280,720,1500,0.0172980638571094,3487.63103678930
10,1.00,1280,720,2350,0.0149937197773119,5381.76334448161
11,0.125,1920,1080,350,0.0392005881613388,1597.77648160535
12,0.250,1920,1080,470,0.0364310657849926,1913.16596655518
13,0.375,1920,1080,630,0.0338722722181303,2357.18993979933
14,0.500,1920,1080,875,0.0311661447071784,2975.12901672241

```

15, 0.625, 1920, 1080, 1130, 0.0279003426232315, 3823.09570568562
 16, 0.750, 1920, 1080, 1520, 0.0239717957468280, 5019.52345150502
 17, 0.875, 1920, 1080, 2040, 0.0203521125447130, 6591.28331772575
 18, 1.000, 1920, 1080, 2750, 0.0177087487784741, 8593.41056856188

C. DASH representations for The Swiss Account (TSA)

```
segmentDuration=2s
numberOfSegments=299
AvgSigma/mu=0.036956395701932
reprId,QualityIndex,screenWidth,screenHeight,bitrate,sigma/mu,avgchunksize
1,0.25,480,360,80,0.0579961265233233,119.833658862876
2,0.50,480,360,165,0.0415561385066449,242.090782608696
3,0.75,480,360,375,0.0340395888285653,554.896000000000
4,1.00,480,360,730,0.0306446772405097,1049.78207357860
5,0.17,1280,720,300,0.0435884614621050,638.184615384615
6,0.33,1280,720,420,0.0386343135836840,847.400133779264
7,0.50,1280,720,585,0.0343886599405611,1132.17717725753
8,0.67,1280,720,920,0.0299019339446362,1717.18362541806
9,0.83,1280,720,1500,0.0266391479526706,2739.08074916388
10,1.00,1280,720,2350,0.0233906396962136,4208.10408026756
11,0.125,1920,1080,350,0.0523025234228145,1087.96709030100
12,0.250,1920,1080,470,0.0474700853112857,1351.75248160535
13,0.375,1920,1080,630,0.0432705501707734,1700.84294314381
14,0.500,1920,1080,875,0.0390027996581480,2163.26156521739
15,0.625,1920,1080,1130,0.0349430973204769,2763.15630769231
16,0.750,1920,1080,1520,0.0305472934277396,3589.04395986622
17,0.875,1920,1080,2040,0.0264937384651144,4651.42793311036
18,1.000,1920,1080,2750,0.0223928047654375,6079.55106354516
```

REFERENCES

- [1] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- [2] Giacomo Calvigioni, Ramon Aparicio-Pardo, Lucile Sassatelli, Jeremie Leguay, Paolo Medagliani, and Stefano Paris. Quality of experience-based routing of video traffic for overlay and isp networks. In *Proc. IEEE Intl. Conf. on Computer Communications (INFOCOM)*, 2018.
- [3] ITU-T. Parametric non-intrusive bitstream assessment of video media streaming quality - Higher resolution application area , 2013.
- [4] Christian Kreuzberger, Daniel Posch, and Hermann Hellwagner. Amust framework - adaptive multimedia streaming simulation framework for ns-3 and ndnsim, 2016.
- [5] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. *SIGCOMM Comput. Commun. Rev.*, 45(4):325–338, August 2015.
- [6] F. Zhang, W. Lin, Z. Chen, and K. N. Ngan. Additive log-logistic model for networked video quality assessment. *IEEE Transactions on Image Processing*, 22(4):1536–1547, April 2013.