

PAPER • OPEN ACCESS

Modeling Media Player Switching between Bit Rates

To cite this article: R V Voronov *et al* 2019 *J. Phys.: Conf. Ser.* **1352** 012060

View the [article online](#) for updates and enhancements.

You may also like

- [A Grain of Truth: The Media, the Public, and Biotechnology](#)
Susanna Hornig Priest
- [Comparative study of the methodologies used for subjective medical image quality assessment](#)
Lucie L  v  que, Meriem Outtas, Hantao Liu et al.
- [Achieving high bit rate logical stochastic resonance in a bistable system by adjusting parameters](#)
Ding-Xin Yang, , Feng-Shou Gu et al.



ECS
The
Electrochemical
Society
Advancing solid state &
electrochemical science & technology

DISCOVER
how sustainability
intersects with
electrochemistry & solid
state science research

Modeling Media Player Switching between Bit Rates

R V Voronov, A A Rogov, A V Brilev and E A Petrov

Petrozavodsk State University, Institute of mathematics and information technology,
Petrozavodsk, Russia

E-mail: rvoronov@petsu.ru

Abstract. The article presents the simulated media player for faster switching between available bit rates with the media stream. Also the investigation includes the results of simulated behavior of the media player while it is receiving media streams in different networks. It is important to identify the real bit rates that are used by the media player for playing media streams at each quantum of time. These results allow better determination of the process of media streams' encoding bit rates which fit best for playing media without delays among the maximum number of users and streams.

1. Introduction

In order to get the right bit rate, it is important to find out what bit rate is required by the users. On this evidence the following approach is suggested. If the page is opened before the stream, then a test media flow with a few available bitrates is started up. The player that is integrated into a web page reports to the remote server what bit rate player is being used at the moment, as well as possible delays in playing of a media flow. On the basis of the information provided the system suggests three most feasible bitrates for the stream. After launching the stream the video player keeps sending the statistics about its performance. On this basis the system may provide recommendations to the IT administrator regarding bitrates for one of the media streams.

2. Empirical data

This experiment uses the test media stream which has nine different bitrates. The video is four seconds long thus it ensures the optimal performance in streaming and encoding. More detailed description of the experiment is given in [1]. Empirical data received in Wi-Fi had shown that bit rate switches either from second 5 to 9, second 13 to 19 or second 21 to 26. Figure 1 demonstrates some bit rate switching over a time scale. The probability of switching on either of three periods was calculated for the first, second and third periods and amounted for 0.846; 0.12 and 0.034 accordingly [2].

“Statistica” programming package was used to determine that switching during each given period was carried out according to the Weibull-Gnedenko model [3]. Variables for the first, second and the third period of time were as following: $a=7,345$ $c=9,01$; $a=15,589$ $c=16,675$; $a=23,637$ $c=22,0219$ respectively. Variable a stands for a scale parameter and variable c stands for a shape parameter. The research was based on the chi-square test with a significance point of 0,05.

The probability of switching the bit rates was estimated for each given period. Table 1 shows the data for the period 1. The first column shows the current bit rate and other columns show the probability of switching according to the bit rate.



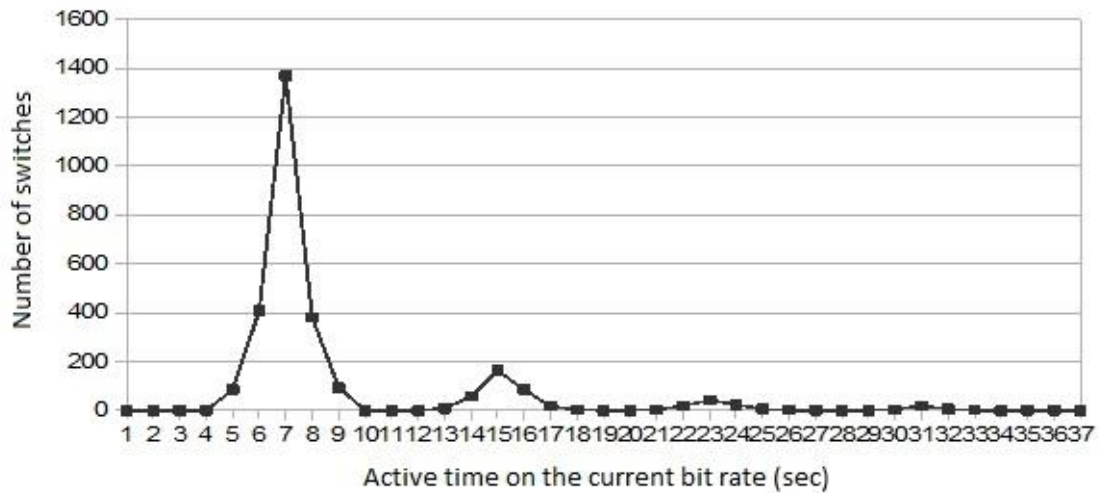


Figure 1. Number of switching to the new bit rate.

Table 1. Probability estimates.

Current bit rate	New bit rate								
	250	500	800	1100	1500	2400	3000	4000	6000
250	0	0	0	0.1675	0.1675	0.33	0.1675	0.1675	0
500	0	0	0	0	0.43	0.43	0.14	0	0
800	0.026	0	0	0.079	0.66	0.05	0.079	0.106	0
1100	0	0	0.0196	0	0.4902	0.255	0.098	0.1372	0
1500	0.0143	0.021	0	0.0357	0	0.236	0.486	0.2	0.007
2400	0.014	0.014	0.007	0.063	0.148	0	0.472	0.282	0
3000	0.0056	0	0.0056	0.0028	0.0746	0.098	0	0.7899	0.0505
4000	0	0	0.0012	0.0023	0.0207	0.0345	0.1253	0	0.816
6000	0.0083	0	0	0.0014	0.011	0.0152	0.1698	0.7943	0

3. Simulation modeling of switching the media player

To solve the optimization task of selecting the bitrates, it is critical to set up a large database. This is why a simulation model was created which emulates the process of changing among available bitrates. This model can simulate the behavior of a player while receiving media in different networks.

Solving the optimization task helps to develop a range of bitrates that ensures good quality of video with minimum delays for as many users as possible. The choice of the bit rate range depends on users' internet connection features[1].

Now we will enter the variable that describes input data of the optimization task. The available bitrates of the media flows are:

$$u = \{u_1, u_2, \dots, u_l\} \quad u_1 < u_2 < \dots < u_l.$$

Then we define the multiple data that contain static information received in the course of media players' behavior prediction:

$$(y_m(t), z_m(t)), \quad m \in \{1, \dots, n\}, \quad t \in \{0, \dots, T\},$$

$$z_m(t) = \begin{cases} 0, & \text{no delays in playing} \\ 1, & \text{delays in playing} \end{cases}$$

n – number of users,

T – time period of receiving the data

$y_m(t)$ – bitrate of the video that is played at the moment (kbit/s)

To define $\bar{y}_m(t)$:

$$\widetilde{y}_m(t) = \begin{cases} y_m(t), & \text{if } z_m(t) = 0; \\ u_{k-1}, & \text{if } z_m(t) = 1 \text{ and } y_m(t) = u_k, k > 1; \\ u_1, & \text{otherwise.} \end{cases} \quad (1)$$

This variable represents the bit rate on which player is able to perform without any delays in real time.

A certain pattern of media player behavior was identified during the experimental data processing obtained in PetrSU (Petrozavodsk State University) wireless network. Frequently after 5 minutes of launching a player was displaying the media flow within 5-10 seconds with the lowest bit rate available. Therefore in order to restrict such fluctuations from affecting the core data, the following limitation was introduced:

$$\text{We find } \mu_m = \sum_{t=0}^T I \left(\widetilde{y}_m(t) - \min_{\tau \in [0, T]} \widetilde{y}_m(\tau) \right) (t), \quad (2)$$

where $I(x) = \begin{cases} 1, & \text{if } x = 0; \\ 0, & \text{otherwise.} \end{cases}$

$$g_m(t) = \begin{cases} \widetilde{y}_m(t), & \text{if } \mu_m \geq f * T \\ u_{k+1}, & \text{if } u_k = \widetilde{y}_m(t) \end{cases} \quad (3)$$

We will enter the variable where f is a threshold value of a possible playing time on the lowest bit rate; in this research $f=0.05$. The variable equals to the bit rate that allows the player to work without delays for not less than $100(1-f)$ percent of the time.

We will set the range of p bit rates $\widetilde{x}_i \in u, i = 1, 2, \dots, p, \widetilde{x}_1 < \widetilde{x}_2 < \dots < \widetilde{x}_p$.

Now we will set the limits that allow using no more than 80% of the network capacity. We assume that C is the maximum capacity of the server and is to be measured in kbit/s. Then we enter variables $c_1(t), c_2(t), \dots, c_p(t)$ - that determine the number of media players that will use bit rates $\widetilde{x}_1, \widetilde{x}_2, \dots, \widetilde{x}_p$ at the moment t , accordingly.

$$c_i(t) = \sum_{t=0}^T I(y_i(t) = x_i). \quad (4)$$

We assume that k is the threshold value of the network capacity of the server, thus $k=0.8$. Therefore we get the limiting condition as following:

$$\widetilde{x}_1 c_1(t) + \widetilde{x}_2 c_2(t) + \dots + \widetilde{x}_p c_p(t) \leq k * C. \quad (5)$$

The solution to the optimization task will be the set $\widetilde{x}_1, \widetilde{x}_2, \widetilde{x}_3$, so that the objective function is minimal:

$$\sum_{t=0}^T \sum_{m=1}^n \min_{1 \leq j \leq 3} \{ (g_m(t) - \widetilde{x}_j), \text{ т. ч. } (g_m(t) - \widetilde{x}_j) \geq 0 \} \rightarrow \min \quad (6)$$

$$\text{If limits (2), (3) are not applied, } g_m(t) = \widetilde{y}_m(t). \quad (7)$$

If a database is rather small, the method of full enumeration can be applied (1)-(6). Otherwise, the dynamic programming method is proposed. Therefore we can highlight three optimization tasks:

1. The first optimization task is described in equations (1), (6), (7).
2. The second optimization task is described in equations (1)-(3), (6).
3. The third optimization task is described in equations (1)-(6).

Limitations (4), (5) should be applied if the number of users is large, hence there is a risk of the network capacity excess.

Then we consider the method of obtaining the input values to determine a set of bit rates. Most of

the latest media players have inline API, which allows receiving the characteristics of a player's performance in real-time.

It is suggested to run an automatic test of users' possibility to obtain the media flow with different bit rates when a translation is offline. Statistic information is sent to the analysis and monitoring system during the playing. On the basis of the received information we get an array $(y_m(t), z_m(t))$, $m \in \{1, \dots, n\}$, $t \in \{0, \dots, T\}$. Similarly, we can receive this information when the playing is paused.

On the basis of the entries array it is possible to select the set of bit rates for the media stream. For the next step, the administrator sets the chosen bit rates in the transcoder settings and launches the main translation. After this the media player switches to the main media stream and continues to send statistics about its performance. However, not all information is available during the playing of the test media flow. In order to fill the missing part, it is strongly suggested to use the imitation model of the player's behavior.

When using the imitation model, it is to be considered that fluctuations in characteristics of the internet connection over a given period will be minor [1].

Statistic information about the process of the media flow is used as input data. It is received during both tests and main media flows. We will determine the input data used for running the imitation model.

The length of the video is Δt (sec).

$i = \{i_1, i_2, \dots, i_r\}$ are periods when switchings between bit rates have occurred $i_1 = (\Delta t - \frac{\Delta t}{2}, \Delta t + \frac{\Delta t}{2}]$, $i_2 = (2\Delta t - \frac{\Delta t}{2}, 2\Delta t + \frac{\Delta t}{2}]$, $i_r = (r\Delta t - \frac{\Delta t}{2}, r\Delta t + \frac{\Delta t}{2}]$.

Intervals when switching between bit rates have occurred?

Now we determine the variable θ_{kt} to estimate the probability that switching from u_k - bit rate $k=1, \dots, l$, to the new bit rate will happen in t - seconds after the start of playing u_k - of the bit rate.

Switching from one bit rate to another in one step can be included into the matrix of the transition probability

- $P(i)$.

$$P(i) = \begin{pmatrix} p_{11}(i) & p_{12}(i) & \dots & p_{1l}(i) \\ p_{21}(i) & p_{22}(i) & \dots & p_{2l}(i) \\ \dots & \dots & \dots & \dots \\ p_{l1}(i) & p_{l2}(i) & \dots & p_{ll}(i) \end{pmatrix}$$

where $p_{kj}(i)$ is the estimate of the probability of switching from the bit rate u_k to the bit rate u_j , during period i . Each line of the matrix in size $l \times l$ shows the estimate of probability from one bit rate to another.

For the entire duration of the user's performance, we determine ϑ - a matrix to estimate the probability of switching between bit rates, the same for each period.

$$\vartheta = \begin{pmatrix} p_{11} & p_{12} & \dots & p_{1l} \\ p_{21} & p_{22} & \dots & p_{2l} \\ \dots & \dots & \dots & \dots \\ p_{l1} & p_{l2} & \dots & p_{ll} \end{pmatrix}$$

Matrixes $P(i)$ and ϑ can be represented via a graph of the switching probabilities.

Apexes indicate possible bit rates and arrows from one apex to another point at possible switches. The graph below (figure 2) is the example of possibilities of switching for ϑ when $l=4$.

Now we determine S_k - the estimates of the delays possibilities when playing with the bit rate u_k .

Then we define the vector of the initial state - $p_k^{(0)}$, the estimate of possibility of using u_k bit rate so that $\sum_{k=1}^l p_k^{(0)} = 1$.

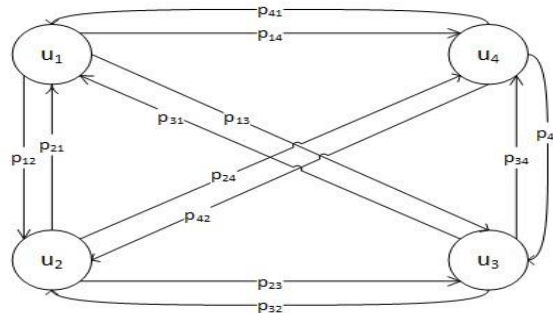


Figure 2. Graph of the possibilities of switching the bit rates.

Now we look into the algorithm of the imitation model's operation. Figure 3 shows an example of the time graph of switching the bit rates during the process of modeling the performance of the media player.

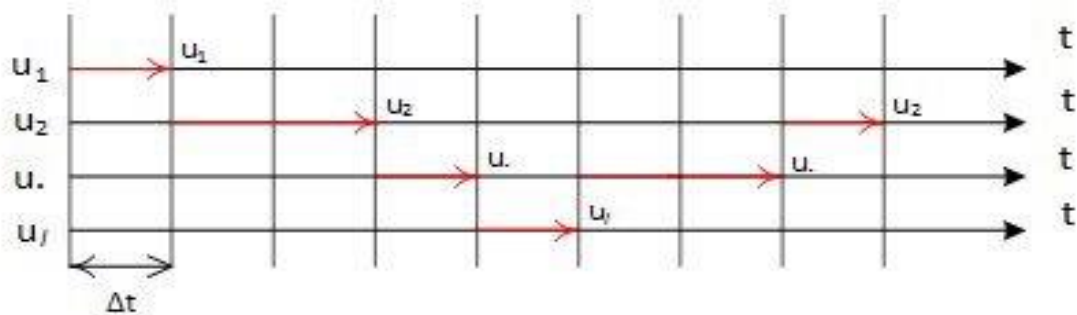


Figure 3. The time graph of switching between the bit rates.

1. During the first stage, the required period is set. The selection of the statistic data as well as the calculation of the input data is carried away for the following steps. (Statistic data are selected and input data are calculated for the following steps.)
2. The selection of the starting bit rate is based on the initial condition – $p_k^{(0)}$.
3. Determination of the period – t is done after the media player switches to another bit rate. This calculation is done based on the estimation θ_{kt} . Then the period i is selected, it means the amount of time during which the switching was carried out. Period i is based on period t .
4. The next step includes setting of the bit rate that the player has been switched on. This value is calculated on the basis of $P(i)$. In case if all the estimations of switching possibilities are zero, then calculation is based on – ϑ .
5. For the next step we need to confirm the duration of the delays at the given time and bit rate. This value is calculated using S_k .
6. The next step is to save the values obtained at steps 2,3,5 into the database that describes the player's performance and to assign the bit rate from step 4 to the initial value.
7. Then we should repeat the steps 3-6 until the accumulated operation time does not exceed or equals the period T .
8. All the steps above should be applied to all users.
9. The system sends back the database which contains information regarding all the media players [4].

We also have developed StreamMon [5] web service, which is designed to organize and control the live-translations as well as monitor the playing process of the end users. This service allows running the automatic test and analysis of the users' capability to receive the media flows via MPEG-DASH in real time with different bit rates [6].

During translation the system provides the administrator with the information about media flows' playing characteristics. The information is provided through the interactive graphs. All the information on the conducted translations is saved into the database and is available for further use.

This service is already implemented in the PetrSU and is used for conducting the live translations.

4. Conclusion

We developed the system of simulation modeling, which can switch the bit rates of the media flows during the live translations. The system can be used for the emulation of the media player's performance if specified initial values are relevant.

It has been established that the bitrates switching happens during one of the three periods. Within each of the periods, switching occurs according to the Weibull-Gnedenko model. The characteristics of allocations for every period are described in the article.

References

- [1] Petrov E A 2016 Modeling of media player switching between bit rates *Moscow: Remote and virtual education* ed E A Petrov and A A Rogov **9 (111)** 48–56
- [2] Zabrowski A L 2013 Criterion for assessing the quality of educational multimedia streams transmitted in real time *Proceedings of Karelian research centre of RAS. Series: Mathematical modeling and information technologies* ed A F Titov **4 1** 26–32
- [3] *Big Data Analytics, Enterprise Analytics, Data Mining Software, Statistical Analysis, Predictive Analytics* <http://www.statsoft.com/> (accessed 22.03.2016)
- [4] *Video Encoding Service for Adaptive Streaming* <https://www.bitcodin.com/> (accessed: 23.03.2016)
- [5] Web service "StreamMon", organization and management of live-broadcasts <http://streammon.petrso.ru> (accessed 22.03.2016)
- [6] *MPEG-DASHLP-VR6 - bitdash HTML5 Player for HLS and MPEG-DASH* <http://www.dash-player.com/> (accessed 22.03.2016)