Online Server-side Optimization Approach for Improving QoE of DASH Clients

Oussama El Marai*† and Tarik Taleb†
*Ecole nationale Superieure d'Informatique, Algiers, Algeria. Email: o_el_marai@esi.dz
†School of Electrical Engineering, Aalto University, Finland. Email: tarik.taleb@aalto.fi

***Abstract*—The many advantages of Dynamic Adaptive streaming over HTTP (DASH) made it one of the most prevalent video streaming technologies in recent years. Unfortunately, many studies have unveiled the QoE issue of users when multiple DASH clients compete for the bandwidth of a bottleneck link. This issue consists of several aspects, namely the frequent encoding changes, the unfair bandwidth allocation, the inefficient bandwidth utilization, and the relatively long convergence time. These aspects are indeed conflicting each other and resolving them entails tradeoffs. In this paper, we propose a new mathematical model that leverages a score matrix to ensure a fair sharing of the server's bottleneck link between competing clients and satisfies the requests of as many clients as possible and that is for efficient bandwidth utilization. The proposed solution is compared against notable solutions through computer-based simulations, and the results show that the proposed solution achieves high scores in terms of both efficiency and fairness.**

I. INTRODUCTION

In recent years, video streaming services have dominated most of the Internet traffic and have become the most popular service. YouTube is a strong indication for the phenomenal success of streaming services. It has over a billion of users and is serving hundreds of millions of hours on a daily basis according to YouTube's official website. This tremendous expansion is supported by both the availability of heterogeneous networks (e.g., cellular and WIFI networks) and the prevalence of smart devices that have facilitated both the creation of video content and the ubiquitous access to an ever-growing library of video content anywhere and anytime.

Several reasons have pushed for imposing Dynamic Adaptive Streaming over HTTP (DASH) as the de-facto technology for video delivery during the last few years. It allows traversing Network Address Translation (NAT) gateways and firewalls, since it uses the ubiquitous HTTP/TCP protocol and dynamically adapts the video quality to the client's conditions, such as device capabilities, buffer state and available bandwidth. To this aim, at the server end, each video is encoded at different bitrates whereby each version is chopped into small chunks with the same time duration. A manifest file is also generated that contains information about the existing versions of the video, such as the URLs of the different chunks and their corresponding timing and bitrate. At the other end, a DASH client continuously assesses its network speed and accordingly selects the appropriate video quality, taking into account its buffer state. When a buffer goes empty, the client encounters video stalls until new segments are received. So far, this approach has been the baseline for delivering video content and has been adopted by many popular content providers (e.g., YouTube, Netflix and Vimeo). In spite of the many advantages of DASH, many studies have revealed stability and fairness issues when multiple clients compete for a bottleneck link. These issues are mainly caused by the ON/OFF behaviour of the clients, which adversely affects the users' quality of experience (QoE).

To cope with these issues, different approaches have been proposed ranging from the client-side to the server side. Client-side solutions are mainly based on either bandwidth estimation [1]–[5] or buffer occupancy [6]–[8]. On the other hand, few solutions have been proposed for implementation at the server side [9], [10]. Most of these studies have focused on the QoE issue that can be negatively affected by many factors, such as the start-up delay, the re-buffering phenomenon, the frequent quality changes and the domination of the shared link by some clients [11]. As these properties conflict each other, the design of a successful DASH-based system can be achieved only by finding trade-offs between these multiple objectives.

In this paper, we propose a new solution, dubbed SO-DASH (Server-side efficiency and fairness Optimization for DASH clients), based on a mathematical model that takes into account both maximizing the efficiency of the DASH-based clients and ensuring fairness among clients competing for a bottleneck link, taking into consideration the difference in their bandwidth capacities. For this purpose, we introduce in our model a weight/score matrix, where we give lower scores to clients playing video content at higher bitrates and vice versa.

The reminder of this paper is organized as follows. Section II gives an overview on related research work. The proposed model is described in Section III. The simulation results are presented and discussed in Section IV. Finally, concluding remarks and future research directions are given in Section V.

II. RELATED WORK

In recent years, DASH has become the predominant video streaming technology used by most commercial streaming services (e.g., YouTube, Vimeo, NetFlix and Akamai). This is thanks to its many offered advantages, such as usage of the ubiquitous HTTP protocol, its ability to traverse NATs and firewalls, and its capability to adapt video quality to clients' conditions. However, recent studies [4], [9], [12]–[14], have shown the limitation of DASH in achieving acceptable level of QoE when clients compete for bandwidth. In this vein, enhancing efficiency, fairness and stability of DASH has become an important research challenge that has attracted the attention of many researchers. In this section, we provide a brief overview on recent research work conducted on DASH, focusing on the multi-player scenario.

FESTIVE [1] is the first algorithm that considered the three objectives (i.e., efficiency, fairness and stability) in multi-

978-1-5090-5019-2/17/$31.00 ©2017 IEEE

player scheme. It is a client-side solution that performs three steps before it decides the next bitrate. In the first step, the harmonic mean of the last 20 chunks bandwidth estimation is calculated to smooth the outliers' values. Next, the delayed update module is called to avoid frequent bitrate switching. Finally, a randomized request scheduling is employed to ensure there is no start time bias. In [4], the authors proposed the PANDA algorithm, mainly aiming at keeping the stability of the clients by employing a conservative upshift rate level and more responsive downshift. Cicco et al. proposed the ELASTIC algorithm, using a feedback-control theory approach, to avoid generating on-off traffic patterns aiming for a fair share of the bottleneck resources between multiple video flows [2]. Similarly, a control-theoretic approach using a PD (Proportional Derivative) controller is employed by Zhou et al. to monitor the buffer dynamics for a smooth rate adaptation [8]. The research work presented in [6] introduces a purely buffer-based approach to determine the bitrate of the next chunk without any bandwidth estimation. In [15], the authors proposed a cooperation-based solution between the clients and the server, dubbed ESTC (Enhancing Server and clienT Cooperation), in order to reach high scores for efficiency, fairness and stability. ESTC also allows a quick and smooth convergence to the fair allocation. It consists of two algorithms. The first is located at the client and operates as a player aiming at controlling the client's efficiency and stability, whereas the second resides at the server aiming at ensuring the fair bandwidth allocation among competing clients using the information received from the client (e.g., requested bitrate and client's bandwidth capacity) and those available at the server (e.g., shared bandwidth capacity and number of connected clients).

In [16], Yin et. al., proposed a mathematical model for router-assisted bandwidth allocation whereby the objective is to maximize fairness among users in the perceived QoE. Zahran et al. proposed the OSCAR algorithm for mobile networks whereby the optimization problem aims at avoiding video stalls while keeping high video quality [17]. In [18], Chiariotti et al. formulated the representations selection problem as a Markov Decision Process and proposed an online Reinforcement Learning-based DASH controller that learns the system dynamics and accordingly selects the representation level that maximizes the long-term reward. To quickly converge to high and stable rewards and make the learning process faster, the controller exploits a parallel learning technique that limits the sub-optimal possible choices. The authors in [19] proposed a Software Defined Networking (SDN) based architecture that dynamically allocates to clients the resources they need in order to maximize the per-client QoE and alleviate the scalability issues. The proposed SDN-DASH exploits the powerful features of the emerging SDN technology to manage the network resources for efficient DASH-based streaming.

In conclusion, there have been many research activities to improve DASH. Most of these research work (except [15]) did not consider the characteristics of client devices (e.g., bandwidth capacity) in the bitrate allocation. As a remedy to this, the present work reflects the clients' bandwidth capacity in the bitrate allocation decision process and prioritizes among clients when the shared bandwidth becomes critical. While the authors' work presented in [15] introduces a heuristic-based solution, this manuscript entails a mathematical model of the problem and solves it using CPLEX.

III. MODEL DESCRIPTION & PROPOSED SOLUTION

A. Optimization Problem

In a previous work [15], the authors proposed a heuristic-based solution, dubbed ESTC, to improve efficiency, fairness and the clients' stability. In this paper, a new mathematical model is designed for the purpose of optimizing both the efficiency and fairness in a multi-player scheme. Let consider a set of *n* clients $C = \{c_0, c_1, ..., c_{n-1}\}$ competing for a single bottleneck link *B*, whereby each client c_i has its own bandwidth capacity. Without any loss of generality, we consider client c_i is an old client in comparison to client c_{i+1}. At the server side, the videos are encoded into *m* different video qualities where $L = \{l_0, l_1, ..., l_{m-1}\}$ represents the set of existing representation levels. Each representation level is chopped into small segments of fixed time duration. At each segment request, a client sends a HTTP GET request $r_i \in L$ to the server and the server picks one and only one representation level for the client. We have:

$$\forall i \in C, \sum_{j \in L} x_{ij} = 1 \tag{1}$$

where x_{ij} is a boolean variable that takes the value 1 if the client i is allocated video quality j; otherwise it takes the value 0. The server should ideally satisfy the clients' requests by giving them the requested level r_i, if the shared bandwidth allows it, or responds with a lower level. Accordingly, we have:

$$\forall i \in C, \sum_{j \in L} l_j x_{ij} \leq r_i \tag{2}$$

To avoid overloading the shared bottleneck and the associated buffering phenomenon, the summation of the different clients' allocated bitrates should not exceed the shared bandwidth B:

$$\sum_{i \in C} \sum_{j \in L} l_j x_{ij} \leq B \tag{3}$$

In case of multiple clients having the same bandwidth capacities and playing videos at the same encoding rates, and when the downgrading of a client's bitrate becomes necessary, the system should give priority to old clients in receiving videos at higher bitrates. This condition can be formulated as follows:

$$\forall i \in C \setminus n-1, \sum_{j \in L} x_{ij} - \sum_{j \in L} x_{(i+1)j} \geq 0 \tag{4}$$

A major limitation of many DASH solutions, proposed in the literature, consists in poor fairness. In this paper, fairness is achieved by allocating to clients with similar bandwidth capacities a quality level close to each other. The problem of poor fairness is mainly caused by two factors: the alternating subscriptions to the bottleneck link due to the ON/OFF periods, and the domination of the first connected clients. For

the former, we propose a solution which prohibits the clients observing a short period of improvement in their estimated throughput due to OFF periods of other clients. It consists of considering the client's bandwidth capacity, instead of the requested encoding rate sent to the server, in the decision process at the server side. For the latter cause, there should be a mechanism that forces the clients playing videos at high quality levels to decrease the bitrates they are using in order to give the others the chance for promoting their video qualities and reaching their fair share. In addition, a priority among the clients should be defined when decreasing the video quality. This can be achieved by targeting last comers first and downgrading their representation level when it becomes necessary. For this aim, we introduce a weight matrix p where the rows represent the clients and the columns represent the available video quality levels. The value of the cell p_{ij} represents the weight/score given to Client i when decreasing its quality level from j to $(j-1)$. The objective function aims at minimizing the summation of the p matrix scores. We accordingly have:

$$Min\left(Z=\sum_{i\in C}\sum_{j\in L}p_{ij}x_{ij}\right) \tag{5}$$

B. Objective Function

We formulate the above-described problem as follows:

$$Min\left(Z=\sum_{i\in C}\sum_{j\in L}p_{ij}x_{ij}\right)$$

s.t:

$$\sum_{i\in C}\sum_{j\in L}l_j x_{ij}\leq B$$

$$\forall i\in C,\sum_{j\in L}x_{ij}=1$$

$$\forall i\in C,\sum_{j\in L}l_j x_{ij}\leq r_i$$

$$\forall i\in C\setminus n-1,\sum_{j\in L}x_{ij}-\sum_{j\in L}x_{(i+1)j}\geq 0$$

where Z ensures at the same time the maximization of the bandwidth utilization and the fairness between the clients having equivalent capacities.

x_{ij} : Binary variables. $x_{ij}\in\{0,1\}$.

B : Shared bandwidth.

C : Set of clients. $i\in\{0,1,2,...,n-1\}$

L : Set of different levels. $j\in\{0,1,2,...,m-1\}$

r_i : Requested level of Client i.

p_{ij} : Represents the score of Client i when it is moved from family f_j to f_{j-1}. The matrix p defines a priority, by attributing scores at each representation level, among the different clients when a level decreasing becomes necessary. As our objective function aims at minimizing the total sum of p values, the scores are attributed in a way that the clients having requested high representation levels get lower values. The rows of p represent the different clients and the columns represent the family classes of the clients. Based on the clients' requests (r_i), the different clients are initially classified into families. The number of family classes is m as we have m representation levels. A client i belongs to a family f_j if $r_i=l_j$. A family class f_j could be empty if no client requests the j^{th} representation level.

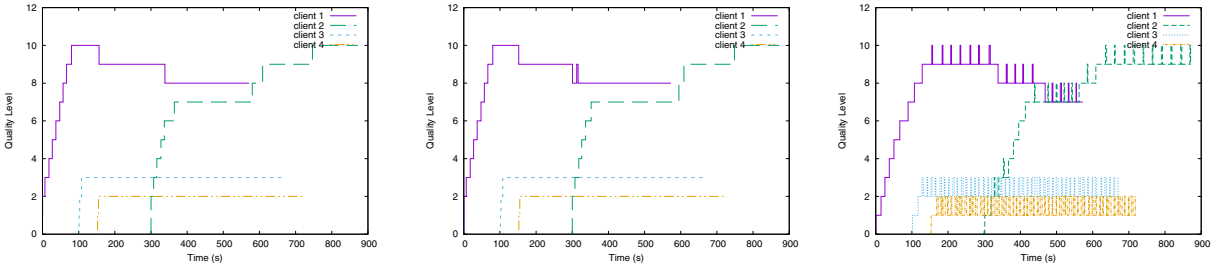
To fill in the p matrix, we first initialize all the matrix by zeros. Then, we start from the last column of the matrix corresponding to the highest representation level, namely j. Suppose that f_j contains the subset of clients α having initially requested the level j. For each client i belonging to f_j, we set p_{ij} to 1, which is the initial score from which we start. This means that 1 is the score given to α members when decreasing their level from j to $(j-1)$. Then, we move to the next lower level which is $(j-1)$. In that level, we have to set at least the clients' scores of α subset if f_{j-1} is empty. Otherwise we should consider the subset β that contains the clients having initially requested the level $(j-1)$ and fill in their scores in the matrix at the column with the index $(j-1)$ before the scores of α at the same column. This means that we give priority to β clients when decreasing the level from $(j-1)$ to $(j-2)$ compared to α clients. To calculate the score of β clients when decreasing the level from $(j-1)$ to $(j-2)$, we just add one to the cumulative amount of all previous scores. For α clients' scores at $(j-1)$ level, we apply the same logic by adding one to the summation of the previously attributed scores. By applying this logic, the program loops over the existing representation levels until it attains the lowest level. The clients' subsets (i.e., α,β,γ...etc) correspond to the family classes where a higher priority (corresponding to lower scores), for decreasing the representation level when different subsets coexist at the same family class, is given to the subsets having initially requested a lower representation level and vice versa. The idea behind is to avoid downgrading clients of f_j from level j to $(j-2)$ without affecting the clients having initially requested $(j-1)$ level, which helps improving the clients' QoE by ensuring a smooth downgrading process. To clarify the p matrix calculation process, we use the following example, considering the following configuration:

- $B=6000$.Kbps
- $L=\{300,700,1000,1500\}$.Kbps
- $C=\{0,1,2,3,4,5,6,7,8\}$.
- $R=\{300,1500,300,1000,700,1000,1000,1500,700\}$.

Initially, the family classes that could be derived from that configuration are the following:

- $f_3=\alpha=\{c_1,c_7\}$.
- $f_2=\beta=\{c_3,c_5,c_6\}$.
- $f_1=\gamma=\{c_4,c_8\}$.
- $f_0=\delta=\{c_0,c_2\}$.

We start by giving the score 1 to α clients, having requested the highest encoding i.e. $1500Kbps$, which is the score given to those clients when reducing their level from l_3 to l_2. After reducing the representation level of α clients, f_3 becomes empty and f_2 will contain $\beta\cup\alpha$. The next step consists of calculating the scores of β clients. It consists of the summation of all matrix values at this stage plus 1. In our case, the new score should be 3. After finishing with β clients, we apply the same process to α clients where we set their cells at l_2



(a) SO-DASH. (b) ESTC. (c) FESTIVE.

Fig. 1: The played representation levels when four clients compete for a shared bandwidth of 10Mbps.

column to 12, f_2 becomes empty and $f_1 = \gamma \cup \beta \cup \alpha$. We repeat this process for the subsets of f_1 (in the given order) and the following family classes until the lowest family class becomes empty. For the previous example, the matrix of the resulting scores is the following:

$$p = \begin{array}{c} \\ c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{array} \begin{array}{c} \begin{array}{cccc} l_0 & l_1 & l_2 & l_3 \end{array} \\ \begin{pmatrix} 1296 & 0 & 0 & 0 \\ 46656 & 432 & 12 & 1 \\ 1296 & 0 & 0 & 0 \\ 11664 & 108 & 3 & 0 \\ 3888 & 36 & 0 & 0 \\ 11664 & 108 & 3 & 0 \\ 11664 & 108 & 3 & 0 \\ 46656 & 432 & 12 & 1 \\ 3888 & 36 & 0 & 0 \end{pmatrix} \end{array}$$

This p matrix will then be used by the proposed model to determine the clients that should be downgraded in case of critical bandwidth.

IV. PERFORMANCE EVALUATION

To evaluate the performance of the proposed model, we conducted many simulations using the NS3 simulator, and compared the results to FESTIVE [1] and ESTC [15]. In the following, we describe the simulation settings under which the simulations were carried out, we then present and discuss the simulation results of 4, 20, 50 and 100 clients, respectively. Note that the evaluation and comparison are based on the metrics described in [15] and the clients' adaptation logic is the same as in [15]. The proposed model is implemented using the Optimization Programming Language (OPL) and resolved with the CPLEX optimizer from IBM.

TABLE I: Simulations Setup.

Number of clients	Shared bandwidth	Starting time
4 clients	10 Mbps	At 0s, 100s, 150s and 300s
20 clients	100 Mbps	Random between 0 and 200s
50 clients	100 Mbps	Random between 0 and 200s
100 clients	300 Mbps	Random between 0 and 300s

A. Simulation Setup

The network topology used in our simulations is a star topology where the server and the clients are all connected to a router. The shared bandwidth capacity (i.e., the link capacity between the server and the router) used in each simulation and the arrival times of the clients are described in Table 1. As to clients' bandwidth capacities, when running 20, 50 and 100 clients, they are set randomly to one of the following values: $10Mbps$, $900Kbps$, $500Kbps$, with the condition that half of the competing clients get $10Mbps$ to create a bottleneck link, ensuring that the summation of the clients' capacities exceeds the shared bandwidth capacity. In the case of running four clients, the clients' capacities are set to: $10Mbps$, $900Kbps$, $500Kbps$ and $10Mbps$, respectively. The set of representation levels used in the simulations is the following: L={30, 100, 400, 800, 1200, 1800, 2200, 3000, 5000, 7000, 9000, 11000}Kbps.

B. Simulation Results

To compare the behavior of the proposed solution against that of FESTIVE and ESTC, we start by running four clients, with different bandwidth capacities and starting at different times, and display the played representation levels by each client. Then, we show how the model behaves at scale by running larger numbers of clients.

1) Comparison of SO-DASH, ESTC and FESTIVE when running four clients: The played representation levels of SO-DASH, ESTC and FESTIVE clients are depicted in Fig. 1. The figure shows that SO-DASH and ESTC achieve a better QoE in terms of stability and convergence time to the fair share, while each client in both solutions play the highest encoding rate which is lower than its bandwidth capacity. From Fig. 1(b), we can see that the connection of client 2 at t=300s caused a level drop, from l_9 to l_8, for client 1 resulting in an instability few seconds later. In case of the SO-DASH solution, we can see from Fig. 1(a) that client 1 behaves better and keeps l_9 until client 2 reaches l_6, resulting in a better bandwidth utilization and avoiding unnecessary level downshift. However, we can see that the convergence of client 2 to its fair share (l_7 at t=364s) in SO-DASH is delayed compared to ESTC (at t=352s).

In terms of efficiency, the comparison between the three solutions in Fig. 2 shows that SO-DASH and ESTC outperform FESTIVE in bandwidth utilization. The latter occasionally achieves non-persistent high scores. We can vividly see that SO-DASH and ESTC exhibit the same performance except within the time intervals [151s,156s] and [300s,338s] where SO-DASH outperforms ESTC (as client 1 in SO-DASH does not decrease its level immediately, unlike ESTC), and the

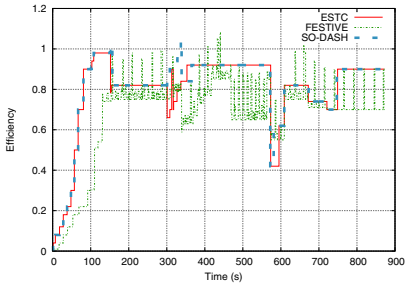


Fig. 2: The efficiency comparison when four clients compete for 10Mbps.

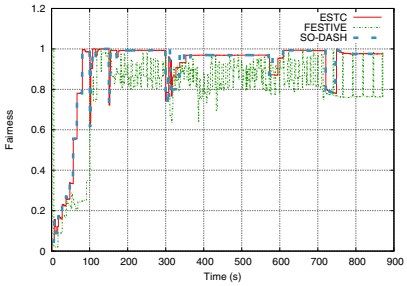


Fig. 3: The fairness comparison when four clients compete for 10Mbps.

Fig. 4: The stability comparison when four clients compete for 10Mbps.

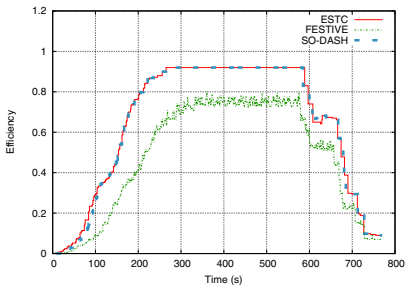


Fig. 5: The efficiency comparison when 20 clients compete for 100 Mbps.

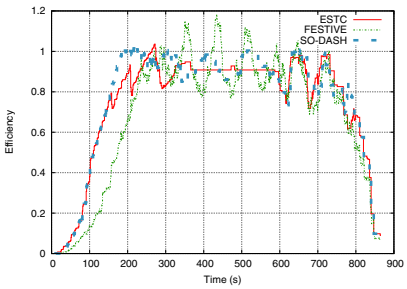


Fig. 6: The efficiency comparison when 50 clients compete for 100Mbps.

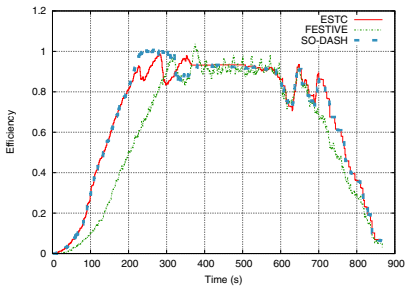


Fig. 7: The efficiency comparison when 100 clients compete for 300Mbps.

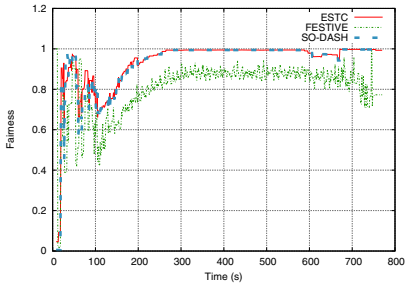


Fig. 8: The fairness comparison when 20 clients compete for 100Mbps.

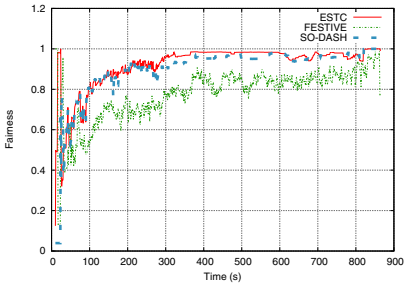


Fig. 9: The fairness comparison when 50 clients compete for 100Mbps.

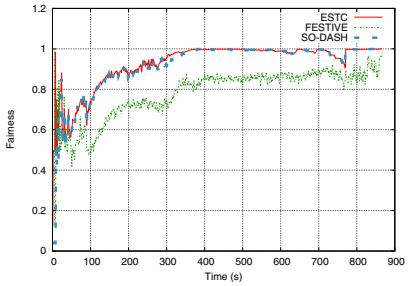


Fig. 10: The fairness comparison when 100 clients compete for 300Mbps.

interval [352s,364s] where ESTC outperforms SO-DASH due to the delayed convergence to the fair share, and between the interval [581s,595s] corresponding to the session end of client1.

With regard to fairness, in Fig. 3, the scores of SO-DASH and ESTC are in general similar with a slight difference. At steady state, the scores achieved are in the vicinity of one. The lower values of fairness occur only when the clients join or leave their sessions. We also note that FESTIVE achieves lower fairness scores compared to SO-DASH and ESTC. This is caused by client 1 that gets a representation level higher than its fair share as depicted in Fig. 1(c). Fig. 4 shows the overall stability metric of the different clients in each solution. It is clear from the figure, and from Fig. 1(c), that FESTIVE is less stable than the two other solutions, while SO-DASH presents a slight stability improvement compared to ESTC.

2) Comparison of SO-DASH, ESTC and FESTIVE when running higher number of clients: In order to evaluate SO-DASH at scale, we conducted simulations with larger numbers of clients, having different bandwidth capacities and starting randomly, competing for the bottleneck link. The results are depicted in Figs. 5-10. For 20 clients, Figs. 5 and 8 show that SO-DASH and ESTC are almost identical either in transient or steady states for both efficiency and fairness metrics. They totally outperform FESTIVE throughout the streaming session.

In the case of 50 clients, Fig. 6 reveals that SO-DASH operates better than ESTC and FESTIVE at transient state, especially during the time interval [150s,250s]. At steady state, SO-DASH and FESTIVE get higher efficiency scores in alternation while ESTC's scores are more stable indicating that the clients do not frequently change their representation levels. Fig. 9 depicts the fairness scores of the three solutions when 50 clients compete for 100 Mbps bottleneck link. It is clear from that figure that SO-DASH and ESTC are more fair in terms of played representation level than FESTIVE, while ESTC is slightly better than SO-DASH. The results shown in Fig. 7 illustrate the efficiency metric values when 100 clients compete for 300 Mbps bottleneck link. At transient state, the previous observation made in case of 50 clients remains true. After stabilizing, the performance of SO-DASH

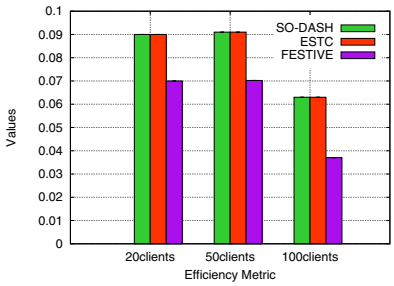


Fig. 11: The overall efficiency metric for 20, 50 and 100 clients.

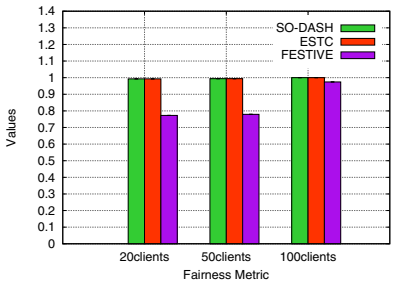


Fig. 12: The overall fairness metric for 20, 50 and 100 clients.

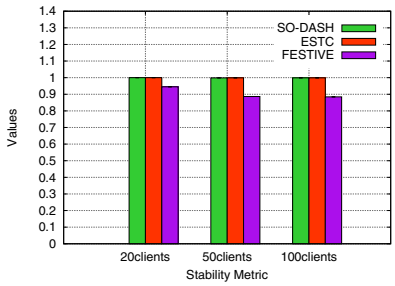


Fig. 13: The overall stability metric for 20, 50 and 100 clients.

and ESTC are almost the same and both are better than FESTIVE which occasionally outperforms them for a short period at steady state. The fairness comparison, depicted in Fig. 10, confirms the observations made with regard to the two previous scenarios.

C. Results Summary

Figs. 11, 12 and 13 summarize the simulations' results of the three solutions in term of overall efficiency, fairness and stability metrics, respectively. From these figures, we can see that SO-DASH exhibits almost the same performance as ESTC for the three metrics with a slight difference, while both ESTC and SO-DASH outperforms FESTIVE. The difference between SO-DASH and ESTC resides at the server's decision approach. The former formulates it as a mathematical optimization problem, whereas the latter employs a heuristically-based approach. The obtained results show that the fundamental observations remain true even in case of larger numbers of clients.

V. CONCLUSION

In DASH, the user's QoE issue is of an outermost importance, especially when the number of competing clients grows. It has many negative aspects with regard to the startup delay, the quick convergence to the fair share, and the maximization of the perceived quality. In this paper, we have proposed a new mathematical model aiming at maximizing the per-client perceived video quality and fairly sharing the bottleneck link. To this aim, we have introduced a weight matrix that allows satisfying the clients' requests which results in a better bandwidth utilization and defines a priority among the clients when the shared bandwidth becomes critical. The results of the conducted simulation show that the proposed model achieves high performance scores and helps improving the user's QoE in terms of perceived video quality and fair bandwidth sharing without adversely affecting the clients' stability. The achieved results are interesting and motivate for future research work that consists of generalizing the proposed model to multiple servers aiming at providing a scalable, QoE-aware and cost-efficient platform using SDN networks.

REFERENCES

[1] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Trans. Netw.*, vol. 22, no. 1, pp. 326–340, Feb. 2014.

[2] 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 20th International*. IEEE, 2013.

[3] C. Liu, I. Bouazizi, and M. Gabbouj, "Segment duration for rate adaptation of adaptive http streaming," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.

[4] 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, April 2014.

[5] T. Taleb, T. Nakamura, and K. Hashimoto, "On supporting handoff management for multi-source video streaming in mobile communication systems," in *LCN 2008. 33rd IEEE Conference on*. IEEE, 2008.

[6] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 187–198, Aug. 2014.

[7] Z. Xu, C. Zhou, L. Liu, X. Zhang, and Z. Guo, "Buffer-based control theoretic approach for dynamically http streaming," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–4.

[8] C. Zhou, C. W. Lin, X. Zhang, and Z. Guo, "Buffer-based smooth rate adaptation for dynamic http streaming," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Oct 2013, pp. 1–9.

[9] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, "Server-based traffic shaping for stabilizing oscillating adaptive streaming players," ser. NOSSDAV '13. New York, NY, USA: ACM, 2013, pp. 19–24.

[10] R. Houdaille and S. Gouache, "Shaping http adaptive streams for a better user experience," in *Proc. of the 3rd Multimedia Systems Conference*. ACM, 2012, pp. 1–9.

[11] T. Taleb and K. Hashimoto, "Ms2: A new real-time multi-source mobile-streaming architecture," *IEEE Trans. on Broadcasting*, vol. 57, no. 3, pp. 662–673, September 2011.

[12] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when http adaptive streaming players compete for bandwidth?" ser. NOSSDAV '12. New York, NY, USA: ACM, 2012, pp. 9–14.

[13] A. Nadembega, A. Hafid, and T. Taleb, "An integrated predictive mobile-oriented bandwidth-reservation framework to support mobile multimedia streaming," *IEEE Trans. on wireless communications*, vol. 13, no. 12, pp. 6863–6875, 2014.

[14] T. Kupka, P. Halvorsen, and C. Griwodz, "Performance of on-off traffic stemming from live adaptive segmented http video streaming," in *37th Annual IEEE Conference on Local Computer Networks*, Oct 2012.

[15] O. ElMarai, T. Taleb, M. Menacer, and M. Koudil, "On improving video streaming efficiency, fairness, stability & convergence time through client-server cooperation," *In Press in IEEE Trans. on Broadcasting.*

[16] X. Yin, M. Bartulović, V. Sekar, and B. Sinopoli, "On the efficiency and fairness of multiplayer http-based adaptive video streaming," *arXiv preprint arXiv:1608.08469*, 2016.

[17] A. H. Zahran, J. Quinlan, D. Raca, C. J. Sreenan, E. Halepovic, R. K. Sinha, R. Jana, and V. Gopalakrishnan, "Oscar: an optimized stall-cautious adaptive bitrate streaming algorithm for mobile networks," in *Proc. of the 8th International Workshop on Mobile Video*. ACM, 2016.

[18] F. Chiariotti, S. D'Aronco, L. Toni, and P. Frossard, "Online learning adaptation strategy for dash clients," in *Proc. of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 8.

[19] A. Bentaleb, A. C. Begen, and R. Zimmermann, "Sdndash: Improving qoe of http adaptive streaming using software defined networking," in *Proc. of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 1296–1305.