Enhancing dynamic adaptive streaming over HTTP for multi-homed users using a Multi-Armed Bandit algorithm

Ali Hodroj, Marc Ibrahim, Yassine Hadjadj-Aoul, Bruno Sericola

HAL Id: hal-02428002
https://inria.hal.science/hal-02428002

Submitted on 4 Jan 2020

Enhancing dynamic adaptive streaming over HTTP for multi-homed users using a Multi-Armed Bandit algorithm

Ali Hodroj, Marc Ibrahim
Saint Joseph University of Beirut
Faculty of Engineering, ESIB
Mar Roukos, Lebanon
ali.hodroj1@net.usj.edu.lb
marc.ibrahim@usj.edu.lb

Yassine Hadjadj-Aoul, Bruno Sericola
Univ Rennes, Inria, CNRS, IRISA
Rennes, France
yassine.hadjadj-aoul@irisa.fr
bruno.sericola@inria.fr

Abstract—Mobile video traffic accounted for more than half of all mobile data traffic over the past two years. Due to the limited bandwidth, users demand for high-quality video streaming becomes a challenge, which could be addressed by exploiting the emerging diversity of access network and adaptive video streaming. In this paper, a network selection algorithm is proposed for Dynamic Adaptive Streaming over HTTP (DASH), the famous international standard on video streaming, to enhance the received video quality to a "multi-homed user" equipped with multiple interfaces. A Multi-Armed Bandit (MAB) heuristic is proposed for a dynamic selection of the best interface at each step. While the Adaptive Bitrate Rules (ABR) used in DASH allow the video player client to dynamically pick the bit rate level according to the perceived network conditions, at each switching step a quality degradation may occur due to the difference in network conditions of the available interfaces. This paper aims to close this gap by (i) designing a MAB algorithm over DASH for a multi-homed user, (ii) evaluating the proposed mechanism through a test-bed implementation, (iii) extending the classic MAB model and (iv) discussing some open issues.

Index Terms—Adaptive, Streaming, Multi-Armed Bandit, Multi-homing.

I. INTRODUCTION

The ever-increasing consumption of video content is one of the most important challenges facing infrastructure providers, as well as content providers and Over-The-Top streaming platforms, such as Netflix. In fact, according to Ciscos mobile traffic forecast [1], the global mobile traffic in 2021 will be doubled compared to 2016, and video traffic will be 82 percent of it .

To deal with such increase in traffic, the Dynamic Adaptive Streaming over HTTP (DASH) protocol [16] was proposed as it allows to react to network fluctuations by adapting the visualized content to the network performance, which increases the users' satisfaction. Indeed, in DASH protocol [15], the video content is splitted into multiple segments with different quality levels described in an XML-based media representation (MPD) file. The DASH client requests the video content at the best quality level that matches with its current available network conditions using HTTP protocol and following an adaptive bitrate streaming rules. This allows different clients to watch same video content with different bit rate according to their throughout and current buffer level [10].

Nowadays, the emerging diversity of access network make most on smartphones possessing multiple network interfaces, which introduce the concept of multihomed devices in heterogeneous networks. Most of mobile users that are multihomed devices may take advantage of this diversity to switch between different networks (Wi-Fi and cellular network) to stream high quality video. However, switching between different networks may degrade the overall video quality as path asymmetry of heterogeneous network have different characteristics (problem 1). Furthermore, due to DASH adaptation strategy, which considers the current throughput and buffer level to choose the quality of next segment, any network switching may impact the advantage of DASH (problem 2).

In this paper, we propose a Multi-Armed Bandit algorithm [14], as network selection mechanism for a DASH multihomed user to switch between Wi-Fi and cellular network for higher quality video streaming, with a new adaptive bitrate rule aiming at reducing the impact of network selection and achieving higher quality. Our solution provides the following features:

- Applying the proposed solution Application layer, which does not require any kernel modification or server side modification.
- Implementing the MAB algorithm as network selection mechanism in DASH player for multihomed user.
- Leveraging the existence of multiple video sources to prevent quality degradation due to traffic congestion at server side. [17]
- Modification of (ABR) rule in DASH to deal with channel diversity.

The rest of this paper is organized as follows: Section II introduces the related work. In Section III we introduce MAB implantatthe ion over DASH. In Sections III we evaluate the performance of MAB in our test bad and address problem 2. Implementation of new ABR rule for Multihomed user is presented in Section IV including a performance evaluation. Finally, Section V concludes this paper and discusses future work.

II. RELATED WORK

The proposed solution falls in the following domains: content delivery networks (CDN) using multi-source, multihoming and multi-path over DASH protocol. Our survey of the similar works shows that –to the best of our knowledge– no solution covers all these aspects together at the application layer. Many solutions were proposed at different protocol layers for

enhancing the concurrent multipath transfer for multimedia content. The related work to this study can be classified into two categories from OSI layer perspective: application and transport layer solutions.

A. Application Layer

In [3], a sub level frame (SFL) scheduling algorithm is proposed. It mainly consists in splitting large-size video frames into sub-frames and dispatches them into different wireless networks using the water filling algorithm, which minimizes the end-to-end delay, and, thus, enhances the video quality for multihomed user.

In [7], a novel loss tolerant bandwidth aggregation approach is proposed to minimize end-to-end video distortion by spreading Forward Error Correction (FEC) packets over a different wireless network during a time interval respecting the delay constraint. In [2], the authors proposed the green bag, as energy bandwidth aggregation middleware for real-time data streaming over asymmetric wireless links. It makes two connections to a media server and each connection requests a partial segment from the video content that is divided into multiple segments, according to their network condition speed and energy consumption.

B. Transport Layer

Multi-Path TCP (MPTCP) is one possible transport protocol to deal with path diversity, it provides a load balancing over different paths to a server by allowing a single data stream to be split across multiple interfaces [8]. ADMIT [9], a quality Driven MPTCP, introduces a reliability-aware flow rate allocation algorithm, to dynamically select the best access network maximizing the video quality, using FEC coding adaption to strike the balance between delay and loss performance to minimize the effective packet loss rate (PLR). Different version of Concurrent Multipath transfer protocol was proposed by the author of [4], [5], [11] and [6] to deliver high streaming video for multihomed mobile user. In [4], the authors proposed a Distortion aware scheme that consists of path status estimation, flow rate allocation and retransmission control to minimize the end to end video distortion rate. In [11], the authors proposed a content aware scheduling strategy by estimating the video content parameters (Frame video distortion level, frame data loss rate and end to end delay of frame) appropriately to schedule the video frame using a decision process at server side to enhance the video quality. In [5], the authors proposed a quality-aware scheme estimating the path quality of each network, using a path quality estimation model (i.e. calculate the RTT of every packet sent on each path), to distribute the data chunks over the qualified path and use an intelligent retransmission policy to handle packet loss problem. In [6], the authors proposed a path failure(PF) detector to classify the different paths into active state or potential failed state. The data are transmitted through multiple paths from the active state list.The paths in potential failed list are monitored using heartbeats packet with an exponential back off of RTO, which

permit them to return to active state list when heartbeat ACK is return.

Both categories allow flows or paths to be established between a client and a single server without taking the advantage of service replication, which is required to prevent server overload problem due to high demand for high quality video. Transport layer solutions require kernel modifications at both server and client site and many network operators do allow MPTCP traffic to pass their middle boxes. Application layer solutions require server side modification without taking the advantages of dynamic adaptive video streaming over HTTP. Therefore, there is a potential to develop a solution to stream high quality videos from multi-sources to multihomed users using the DASH protocol.

III. MULTI-ARMED BANDIT INTERFACE SELECTION STRATEGY

The aim of this work is to propose a network selection algorithm that utilizes the diversity of access networks to maximize the quality of video contents over the HTTP protocol. In Dynamic Adaptive streaming over HTTP, the video content is stored into segments at different qualities/bit rates, known as representations. Segment information (timing, structure, format, quality) are presented in a Media Presentation Description (MPD) file that is downloaded by the client to start viewing the video. An adaptive bit rate algorithm is used to allow the client to switch between video qualities in case of network condition variations according to a set of Adaptive bitrate (ABR) rules queries before every new segment download. These rules are normally divided into primary rules and secondary rules. The primary rules focus on throughput (Throughput Rule) or current buffer status rule (Bola Rule) to determine the quality of the requested segment. The secondary rules avoid rebuffering (Insufficient Buffer Rule), rapid bitrate oscillations (Switch History Rule) and dropped frame due to CPU limitation (Dropped Frame Rule) [18].

Our proposed mechanism is illustrated by considering a mobile DASH client requesting a video content stored at a single server or replicated at different servers. The client through this mechanism is capable of switching to the best network and best video quality. Selecting or switching to the best interface is achieved by implementing a Multi-Armed bandit model on DASH protocol at the DASH Client side.

A. Model

- t_C: The chunk video duration.
- D: The total video duration of the video.
- C: The total number of chunks, which is equal to $\lceil \frac{D}{t_C} \rceil$.
- k: The time step for the multi-armed bandit algorithm. The duration of the time step is equal to $c \times t_C$, where c is the fixed number of chunks to download in a time step.
- $\mathcal{I} = \{I_i\}_{i=1..N_I}$: The set of available N_I network interfaces.
- $d(k)$: The selected interface at time step k (i.e. decision).

- $r(k)$: The reward received at time step k, which is the average bit rate measured at interface $d(k)$.
- $\mathcal{Q} = \{q_j\}_{j=1..N_Q}$ is the set of available N_Q video qualities.

B. MAB Algorithm

The main objective of the MAB algorithm is to select the interface that allows the highest reward and therefore the best quality of experience.

In this work, we will use the ϵ-greedy algorithm that represents a good trade off between exploration and exploitation. In order to select the best interfaces, the algorithm keeps track of the cumulative average reward $\overline{r}_i$ of each interface i in $\mathcal{I}$, that is given by:

$$\overline{r_i} = \alpha r(k) + (1 - \alpha)\overline{r_i}$$

The MAD algorithm selects at each time step k, the interface having the best score with a probability $1 - \epsilon$ (exploitation), or any other interface with probability ϵ (exploration) (see lines 6 to 9). In the latter case, a discover phase is allowed by selecting a random interface, using the $getRandInterface(\mathcal{I})$. As the learning progresses, the ϵ parameter could be decreased to reduce exploration.

For the selected interface i, the function $getNChunks(c, q_{init}, i)$ is used to download c chunks, q_{init} being the quality of the first chunk. The qualities for subsequent chunks will be adaptively determined by the ABR algorithm. The funtion $getNChunks(c, q_{init}, i)$ returns the reward, which is the average bit rate measured at interface i, and the average measured quality.

Algorithm 1 MAB algorithm

1: $q_{init} = DEFAULT$
2: **for** $i = 1$ to N_I **do**   ▷ bootstrap
3:     $\overline{r}_i, q_{init} = getNChunks(c, q_{init}, i)$
4: $k = N_I + 1$
5: **while** $k < \frac{C}{c}$ **do**   ▷ while video not finished
6:     $I^* = \text{argmax}_{i=1..N_I}\overline{r}_i$
7:     **if** $rand() < 1 - \epsilon$ **then**
8:         $d(k) = I^*$ ▷ select the best interface
9:     **else**
10:         $d(k) = getRandInterface(\mathcal{I})$ ▷ random interface
11:     $r(k), q_{init} = getNChunks(c, q_{init}, d(k))$
12:     $\overline{r}_{d(k)} = \alpha r(k) + (1 - \alpha)\overline{r}_{d(k)}$
13:     $k = k + 1$

In this first algorithm, we use the default ABR mode where the next chunk quality is determined by the current measured quality regardless of the interface that will be used to download the next chunk.

C. MAB algorithm with ABR modification

Theoretically, as the quality of the requested segment is determined based on the current status of buffer level and throughput of previous segments according to the ARB lists used in DASH, this may lead to quality degradation in some switching cases when we have different network conditions between the different interfaces. To solve this problem, in any switching phase, we propose to set the quality of first segment requested to last quality detected on same interface as mentioned in Algorithm 2.

Algorithm 2 MAB algorithm with Abr modification

1: **for** $i = 1$ to N_I **do**   ▷ bootstrap
2:     $\overline{r}_i, q_{init}(i) = getNChunks(c, DEFAULT, i)$
3: $k = N_I + 1$
4: **while** $k < \frac{C}{c}$ **do**   ▷ while video not finished
5:     $I^* = \text{argmax}_{i=1..N_I}\overline{r}_i$
6:     **if** $rand() < 1 - \epsilon$ **then**
7:         $d(k) = I^*$ ▷ select the best interface
8:     **else**
9:         $d(k) = getRandInterface(\mathcal{I})$ ▷ random interface
10:     $r(k), q_{init}(d(k)) = getNChunks(c, q_{init}(d(k)), d(k))$
11:     $\overline{r}_{d(k)} = \alpha r(k) + (1 - \alpha)\overline{r}_{d(k)}$
12:     $k = k + 1$

We evaluate the performance of MAB over DASH with and without this modification in the next section.

IV. MODEL EVALUATION

In order to evaluate the performance of our proposed algorithm, we implemented a testbed to avoid the dynamicity of the Internet and allows to regenerate the same configuration settings for testing which permit us to study the performance without any side effect due to network variation metrics between different scenarios. Wondershaper is used to allow the user to limit the bandwidth of one or more interface. The first scenario is considered as a reference scenario (Figure 1) where a standard DASH client watches a video of 10 min with 10 available qualities. The same scenario is repeated with a set of different network speeds(same value for download and upload speed);

Video Bit rate (kbps)	254, 507, 760, 1013,1255,1884,3134,4953,9915,14932
Network Speed (Kbps)	256, 512, 712, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8172, 9196, 10240, 12288, 14336, 16384, 18432, 20480

Fig. 1. Reference scenario parameters

For each scenarios, we calculate the average video rate which represents the average quality perceived according to current network conditions speed. The result shows that better quality could be achieved with fast network speed (Figure 2) and permit us to precise the quality level at each network throughput's level.
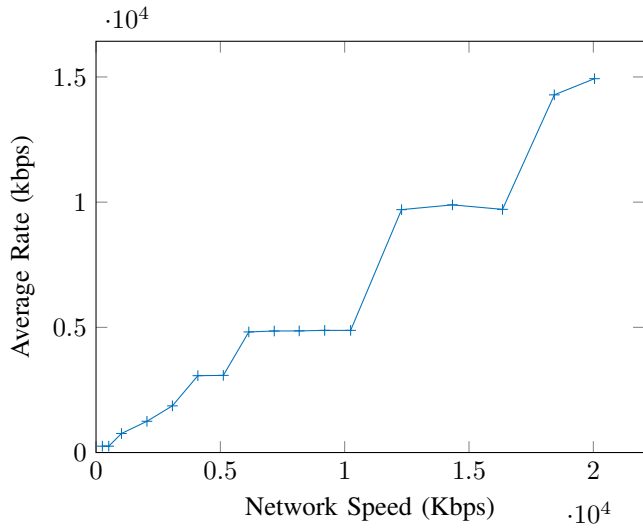
Fig. 2. Video quality variation in function of network speed variation

Fig. 4. Experimental environment

However, Figure 3 shows empty buffer occurrences with different duration according to the network speed variation and available video quality.
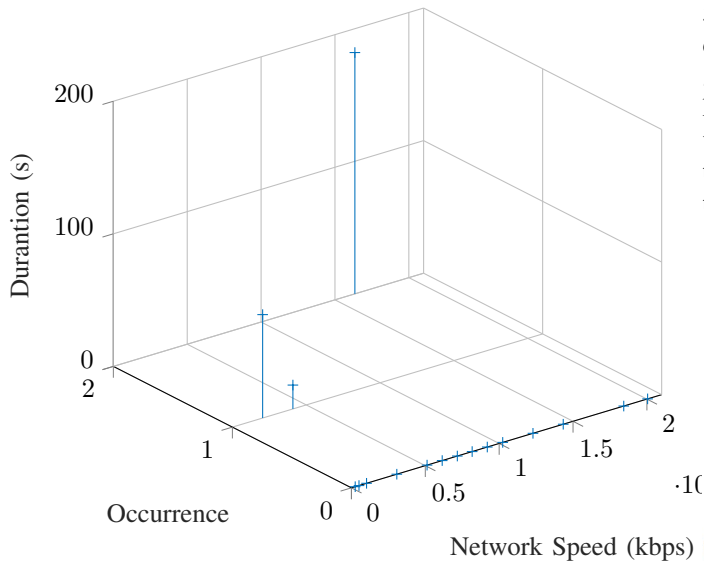

Fig. 3. Duration of empty buffer occurrences as a function of network speed variation

A. Model evaluation

The second measurement set (Figure 4) consists of a multihomed DASH plugged simultaneously to two network connections and two video servers where each server is connected to one of the two networks.The same video content of 10 min with 10 different qualities, is stored at the two servers. MAB algorithm is implemented in DASH at the client side as network selection mechanism without abr modification.
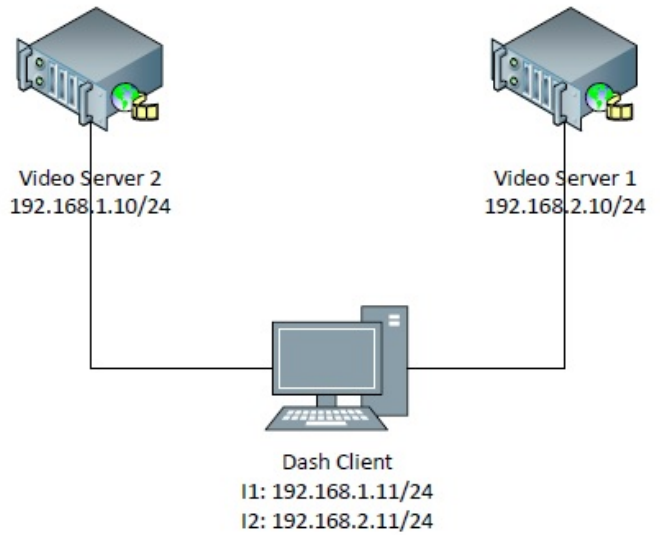
In order to study the impact of eps parameters on the model performance, 81 scenarios were built and run (Figure 5): each scenario consists of watching the video file with different speeds for network connection two (varies from S_0 to S_i) while fixing the speed of first connection to 5120 kbps. The set of chosen speeds, is determined based on those who affect/change the average video bit rate according to the results of first scenario. Then each scenario is repeated 9 times using nine different epsilon values.

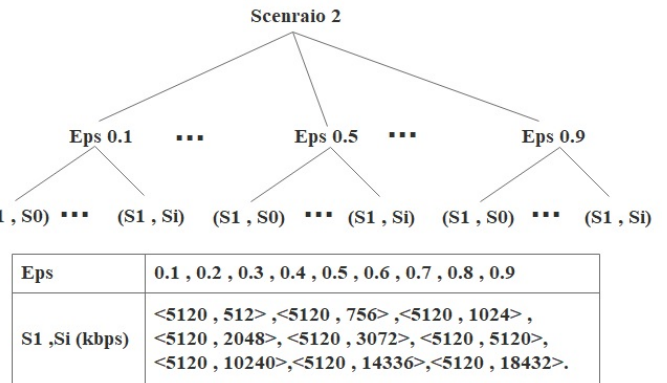| Eps	0.1 , 0.2 , 0.3 , 0.4 , 0.5 , 0.6 , 0.7 , 0.8 , 0.9
S1 ,Si (kbps)	<5120 , 512> ,<5120 , 756> ,<5120 , 1024> , <5120 , 2048>, <5120 , 3072>, <5120 , 5120>, <5120 , 10240>,<5120 , 14336>,<5120 , 18432>.

Fig. 5. Measurement Set 1

In Figure 6, the received video quality is compared between different values of eps. The results show that best average video quality is achieved for 4980 Kbps while eps is set to 0.1 and decreased to 3180 Kbps while eps increased to 0.9.
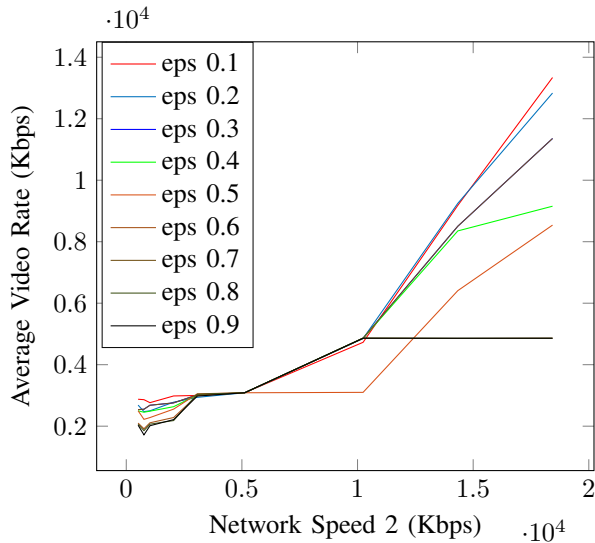
Fig. 6. Average Video quality vs eps value

The video quality is also evaluated based on the presence of empty buffer occurrence with different duration's. Figure 7 shows that a minimum re-buffering duration of 5 seconds appears one time when eps is set 0.1.
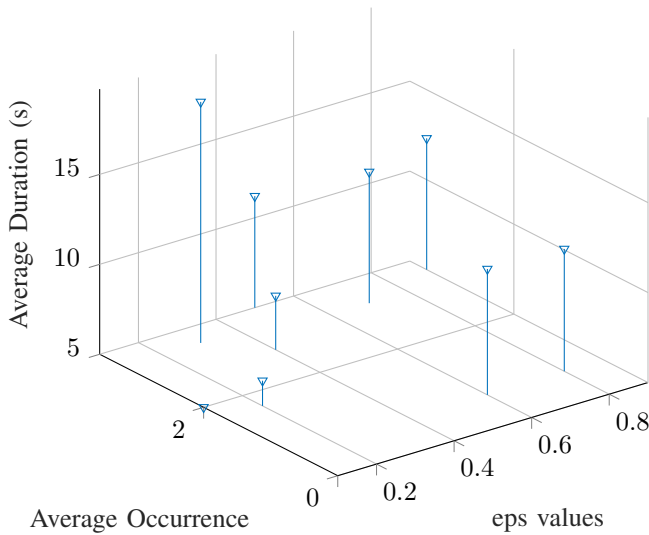

Fig. 7. Average empty buffer occurrence and duration in function of eps variation

According to the results of measurement set 1, we fixed the value of eps to 0.1 and a new set of measurements were run as shown in Figure 8 using 81 different combination values of network speeds.

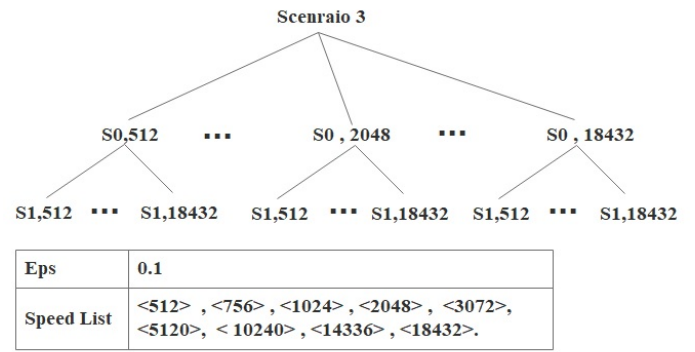| Eps	0.1
Speed List	<512> , <756> , <1024> , <2048> , <3072>, <5120>, < 10240> ,<14336> , <18432>.

Fig. 8. Measurement Set 2 when MAB model is implemented at Dash client without abr modification

In order to compare the performance of MAB model, as network selection algorithm suitable for Dash multihomed client, the bitrate enhancement E (Equation 1) is calculated by measuring the average bit rate Q_i when using network connection i on each scenario.

$$E = \frac{(E_1 + E_2)}{2} = \frac{\frac{Q_1 - Q_{MAB}}{Q_1} + \frac{Q_2 - Q_{MAB}}{Q_2}}{2} \tag{1}$$

Figure 9 illustrates the Cumulative Distributed Function (CDF) of the enhancement metric estimated from the performed scenarios when compared to standard Dash player using each one of these two interfaces alone and to MAB proposed without abr modification . The comparison shows that more than 70% of the scenarios were subject to an enhancement between 20% to 80%. The rest of scenarios still witness an enhancement between 17% and 20% except of 4 scenarios have better performance when using single interface.
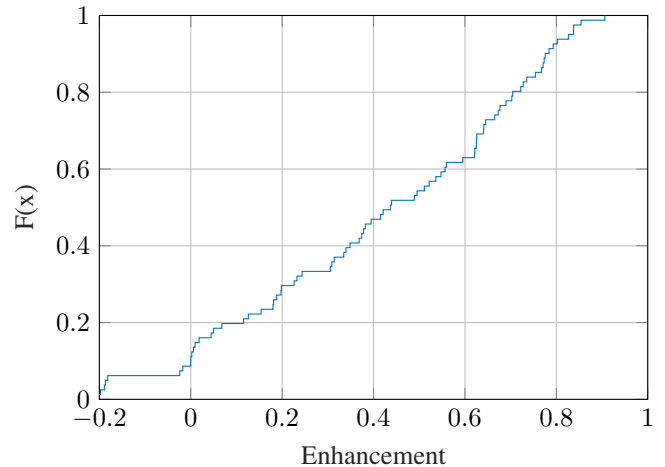

Fig. 9. Average Bitrate Enhancement of MAB-without abr modification

But the excepted theoretically problem in (III.C) was clearly observed in Figure 10 when empty buffer appears in 42% in the testing scenarios with average duration of 3 seconds.
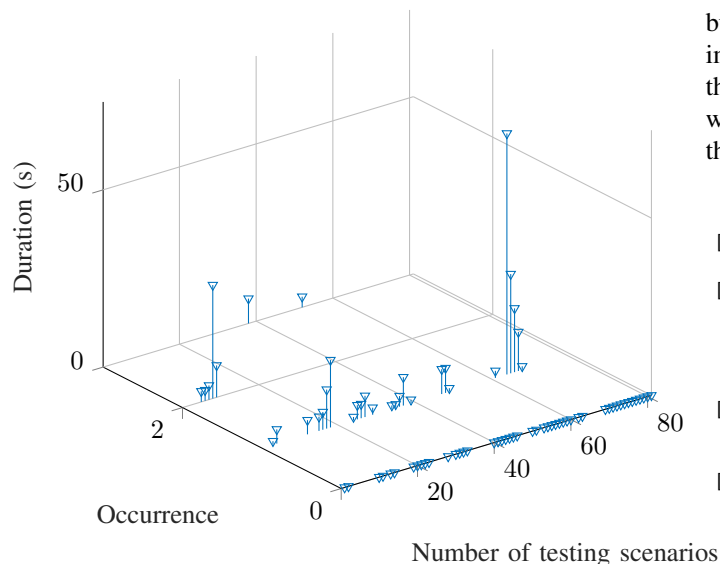
Fig. 10. Empty Buffer Occurrence and Duration of MAB-without modification

The same measurement set 2 scenarios are repeated after modifying the abr rule in DASH in order to deal quality degradation at each switching step due to different network speeds appear between different connections . The results (Figure 10) show absence of any empty buffer occurrence and 80% of the scenarios were subject to an enhancement between 20% to 85%.
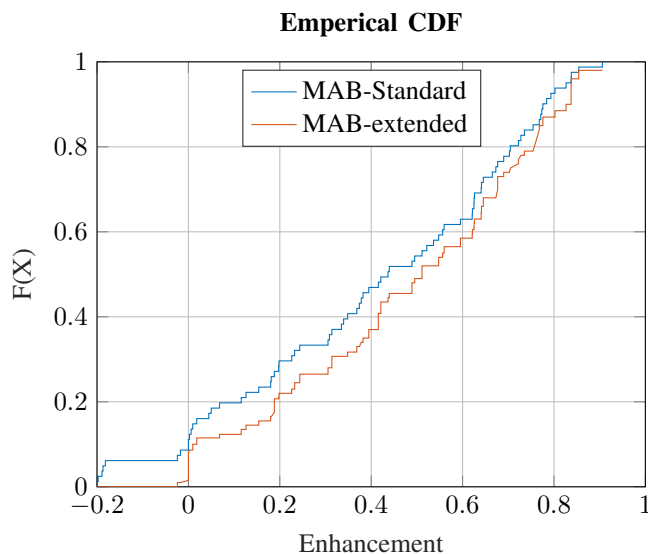

Fig. 11. Average Bitrate Enhancement of MAB-without/with abr modification

V. CONCLUSIONS

In this paper, the MAB algorithm was proposed as network selection algorithm for Multihomed DASH client. We showed that MAB algorithm improves the average video quality but increases the empty buffer occurrences. Then, we modify the abr rule in DASH in order to solve the problem of empty buffer occurrences when using MAB algorithm. Our solution improves the video quality received taking in consideration the dynamicity of different networks used. For future work, we plan to study the security aspects for DASH client taking the advantages of Multihoming.

REFERENCES

[1] (2017). Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021.
[2] Duc Hoang Bui ; Kilho Lee ; Sangeun Oh ; Insik Shin ; Hyojeong Shin ; Honguk Woo ; Daehyun Ban. (2013). GreenBag: Energy-Efficient Bandwidth Aggregation for Real-Time Streaming in Heterogeneous Mobile Wireless Networks. IEEE 34th Real-Time Systems Symposium, 1052-8725.
[3] Jiyan Wu . (2013). A low latency scheduling approach for high definition video streaming over heterogeneous wireless networks. 2013 IEEE Global Communications Conference (GLOBECOM), 1723 - 1729.
[4] Jiyan Wu ; Bo Cheng ; Chau Yuen ; Yanlei Shang ; Junliang Chen. (2015). Distortion-Aware Concurrent Multipath Transfer for Mobile Video Streaming in Heterogeneous Wireless Networks. IEEE Transactions on Mobile Computing, 688 - 701.
[5] Jiyan Wu ; Bo Cheng ; Ming Wang ; Junliang Chen. (2017). Quality-Aware Energy Optimization in Wireless Video Communication With Multipath TCP. IEEE/ACM Transactions on Networking, 2701 - 2718.
[6] Jiyan Wu ; Chau Yuen ; Bo Cheng ; Ming Wang ; Junliang Chen. (2016). Streaming High-Quality Mobile Video with Multipath TCP in Heterogeneous Wireless Networks. IEEE Transactions on Mobile Computing, 2345 - 2361.
[7] Jiyan Wu ; Jingqi Yang ; Junliang Chen. (2013). Loss Tolerant Bandwidth Aggregation for multihomed video streaming over heterogeneous wireless networks. 2013 IEEE Global Communications Conference (GLOBECOM), 2956 - 2962.
[8] Kim ; H. A.; Oh; B. H.; Lee J. (2012). Improvement of MPTCP Performance in heterogeneous network using packet scheduling mechanism. 2012 18Th Asia-Pacific Conference on Communications (APCC), 842 - 847.
[9] Simone Ferlin ; Thomas Dreibholz ; zg Alay. (2014). Multi-path transport over heterogeneous wireless networks: Does it really pay off? 2014 IEEE Global Communications Conference, 4807 - 4813.
[10] Tim Casey ; Gabriel-Miro Muntean. (2016). MPEG-DASH-based framework for improving end-user video experience in heterogeneous multi-network wireless environments. 2016 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), 2345-2361.
[11] Wu J. ; Yuen C. ; Wang M. ; Chen J. (2016). Content-Aware Concurrent Multipath Transfer for High-Definition Video Streaming over Heterogeneous Wireless Networks. IEEE Transactions on Parallel and Distributed Systems, 710 - 723.
[12] Yung-Chih Chen ; Don Towsley ; Ramin Khalili. (2016). MSPlayer: Multi-Source and Multi-Path Video Streaming. IEEE Journal on Selected Areas in Communications, 2198 - 2206.
[13] Vijay Kumar Adhikari ; Yang Guo ; Fang Hao ; Matteo Varvello ; Volker Hilt ; Moritz Steiner ; Zhi-Li Zhang. (2012). Unreeling netflix: Understanding and improving multi-CDN movie delivery. 2012 Proceedings IEEE INFOCOM, 1620 - 1628.
[14] Stefano Boldrini, Luca De Nardis , Giuseppe Caso , Mai T. P. Jocelyn Fiorina and Maria-Gabriella Di Benedetto. (n.d.). muMAB: A Multi-Armed Bandit Model for Wireless Network Selection. Algorithms 2018.
[15] Christopher Mueller ; Stefan Lederer ; Christian Timmerer ; Hermann Hellwagner. (2013). Dynamic Adaptive Streaming over HTTP/2.0. 2013 IEEE International Conference on Multimedia and Expo (ICME), 1 - 6.
[16] Information technology. (2014). Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats. ISO/IEC 23009-1.
[17] Vijay Kumar Adhikari ; Yang Guo ; Fang Hao ; Matteo Varvello ; Volker Hilt ; Moritz Steiner ; Zhi-Li Zhang. (2012). Unreeling netflix: Understanding and improving multi-CDN movie delivery. 2012 Proceedings IEEE INFOCOM, 1620 - 1628.
[18] Kevin Spiteri ; Ramesh Sitaraman; and Daniel Sparacio.(2018) From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player.In MMSys18: 9th ACM Multimedia Systems Conference, 12 - 15. https://doi.org/10.1145/3204949.3204953