Multi-armed bandit algorithms over DASH for multihomed client

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

HAL Id: hal-03509820
https://inria.hal.science/hal-03509820

Multi-Armed bandit Algorithms Over DASH for Multihomed Client

Ali Hodroj*, Marc Ibrahim*, Yassine Hadjadj-Aoul•, and Bruno Sericola•

* Faculty of Engineering, Saint Joseph University of Beirut, ESIB Mar Roukos, Lebanon
• Inria, Univ Rennes, Rennes, France

Abstract—There is an expanding request from the mobile clients of video traffic which has come to over the most recent two years the greater part of all the mobile data traffic. The challenge of providing the clients excellent video streaming is expected principally to the constrained transfer speed which makes it important to misuse the rising decent variety of mobile video spilling and get to systems. For enhancing the video quality received by "multi-homed client" (i.e., clients with multiple network interfaces), a network selection algorithm based on the Multi-Armed Bandit heuristic is proposed on top of the most broadly utilized standard for video streaming Dynamic Adaptive Streaming over HTTP (DASH). For choosing the best network at each progression powerfully, DASH gives mobile video quality dependent on the apparent exhibition from the pre-owned system association through the Adaptive Bitrate Rules (ABR) without investigating the system conditions through the other network(s) which could give better quality. Subsequently, a few alterations for DASH ABR is required to enhance the video quality. Two of the MAB algorithms (UCB and Epsilon Greedy) were embraced for improving MPEG-Dash. The investigations are performed through a proving ground execution to show that UCB surpasses Epsilon Greedy, in stable system conditions, regarding goodput received by the Dash customer. Additionally, UCB can discover the harmony between investigating new choices, and abusing the triumphant variation.

Index Terms—DASH, Multi-homed, video streaming .

I. INTRODUCTION

Mobile traffic will be multiplied in the coming scarcely any years contrasted with 2016 while 82 percent of the information traffic will be for video streaming as indicated by Cisco forecasting [7]. The boundless utilization of smartphones and the developing mainstream video stages have contributed to this remarkable development. Such expanding utilization of video applications is trying from various outlooks from the framework suppliers to the substance suppliers and not least the spilling stages (e.g., Netflix). To manage the variances of the system execution, the Dynamic Adaptive Streaming over HTTP (DASH) convention [18] was proposed to adjust the imagined substance to the system execution, so as to upgrade the nature of experience (QoE) received by the end clients. Fundamentally, the video content is made out of numerous segments having diverse quality levels which are portrayed in an a media presentation description (MPD) record. The DASH customer demands, through HTTP protocol, the video segments with different quality levels that fit with the present status of network conditions and adaptive bitrate streaming rules [15].The outcomes are that dispersed network clients will watch similar video content from a similar server in various quality relying upon end-to-end throughput on the comparing system ways with the server and the present cushion level [10]. Then again, the decent variety of access network interfaces (e.g., Wi-Fi and cell arrange) on smartphones these days, gives such multihomed gadgets the ability to change

starting with one network connection then onto the next dependent on the current network conditions level and to improve the video streaming quality. Nevertheless, there are two issues that can be distinguished in the plan. First issue is that the way asymmetry of heterogeneous network can have various attributes and for this situation exchanging between the various networks may debase the general video quality. The subsequent issue is that the network exchanging could negatively affect the mobile technique of DASH which thinks about the present cradle level and throughput to choose the nature of the following segment. Therefore, we suggest in this article an adaptive network selection algorithms based on the Multi-Armed Bandit algorithms which permit a multihomed Dash use to switch between different networks (WiFi, 3G, and 4G). This solution aims to provide a new adaptive bitrate rule that aims to reduce the effect of network selection and achieve better video streaming quality perceived by the mobile users. This 19 is achieved via the following functions of the proposed solution:

- Applying it at the application level which does not require any modification at the device kernel or server sides.
- Implementing the MAB algorithms in the DASH player of multihomed users which affords an adaptive network selection mechanism that takes benefit of the diversity of the network.
- Leveraging the availability of a couple of resources as video streaming mirror servers to address the excellent degradation that could be caused by the visitors congestion at server side. [17]
- Adjusting the DASH ABR rules to address the channel diversity.

This journal is prepared as follows. Section II provides related studies. Then, our proposed fashions for MAB implementation over DASH is elaborated in Section III. Section IV discusses the measurement scenarios and evaluates our models. Finally, realization and future works are presented in Section V.

II. RELATED WORK

Given that our work falls in the domain of multihoming , multipathing and multi-source over DASH protocol in Content Delivery Networks (CDN), there is no contribution covering every one of these angles together at the application layer. Nonetheless, many arrangements were proposed at the different layers for the presentation improvement of video streaming applications received by the multihomed clients through the simultaneous multipath transmission of media content. We present in this area such commitments in the wake of arranging them into two classes which are application and transport layer arrangements.

A. Application Layer

Sub-frame level (SFL) scheduling approach is introduced in [11] which splits large-size video frames to optimize the delay performance of HD video streaming over heterogeneous wireless networks. It uses a water-filling algorithm for HD video frame splitting and streaming rate allocation over multiple wireless access networks. Also it uses a FEC coding to mitigate wireless channel losses taking the advantage of the flow rate allocation algorithm. This work was the first work to perform the frame-level data protection and transmission of delay-stringent HD video streaming in heterogeneous wireless networks with multi-homed terminals.

In [6], MSPlayer is proposed as a client-based video streaming ,that doesn't require any alteration at TCP stack,and exploits various video sources just as network paths through various interfaces. MSPlayer intends to decrease start-up latency and provide high-quality video streaming. MSPlayer streams videos from multiple YouTube video servers through two interfaces (e.g., WiFi and LTE) at the same. They thought about the exhibition of standard YouTube players and MSPlayer dependent on pre-buffering and re-buffering stages. MSPlayer outperforms both single-path TCP over WiFi and LTE for different specified amounts of pre-buffered video data. Additionally, MSPlayer decreases the start-up delay by 12%, 21%, 28%, individually, for 20/40/60 second buffers.Nonetheless, they don't report the outcomes on how MSPlayer gives heartiness video conveyance in mobile situations.

The author of [14] propose an application-layer transmission framework based on a new Loss Tolerant Bandwidth Aggregation (LTBA) scheme. LTBA consist of is spreading the FEC packets for overcoming the burst channel losses to minimize end-to-end video distortion and delay. The results show that LTBA enhances the average video received quality and guarantee that the video frames are delivered on time even in high loss wireless channel.

GreenBag [3] is an energy-efficient bandwidth aggregation middleware that supports real-time data-streaming services over asymmetric wireless links, requiring no modifications to the existing Internet infrastructure and servers. GreenBag is devised as a middleware operating on the mobile device, requiring neither a proxy server nor any modification to the existing Internet infrastructure and servers. It provides energy-efficient bandwidth aggregation service for real-time video streaming over LTE and Wi-Fi interfaces, adapting dynamically to network fluctuations. It is a multi-link data streaming middleware for supporting reliable and energy efficient real-time data-streaming services, adapting dynamically to varying network conditions. GreenBag aims to provide an optimal mode switch between single-link and dual-link modes, in terms of minimizing energy consumption while meeting the required Quality of Experience (QoE).

B. Transport Layer

A well-known transport layer protocol for multipath bandwidth aggregation is Multi-Path TCP (MPTCP) which works on balancing the load over the different network paths between the server and the user's network connections by splitting the data stream across the multiple network interfaces [4]. It consists of multiple independent TCP connections (subflows). The concurrent use of these links for a TCP/IP session would improve resource usage within the network and, thus improve the user experience through higher throughput and improved resilience to network failure. However, MTCP has a problem that quality leads to degrading due to different characteristics (bandwidth, latency, packet loss rate, etc.) of available asymmetric paths which cause packet out-of-order problem. To solve this problem many intelligent scheduling algorithms at the sender have been proposed to reduce the number of disordered packets. [9] [5].

Kim [23] proposed a scheduling algorithm at the sender to reduce the receive buffer. The main idea is to estimate packet arrival time, by calculating the time that every path to transmit data packet before the transmission of a data packet and schedule packets accordingly. The sender maintains a per path timetable including calculated values of receiving time at the receiver from now for each packet. When the sender has the opportunity to send a new packet, it chooses the path owned smallest transmission time to transmit this packet.

The authors in [1], [19], [22]–[24] propose different transport level solutions for concurrent multipath video streaming for multihoming users. In [23], Jiyan Wu propose *ADMIT* a quality-driven Multi-path TCP that use Forward Error Correction (FEC) coding and rate allocation to reach best quality for real-time streaming. ADMIT uses a rate allocation algorithm to select the appropriate access network and then uses a FEC coding adaptation scheme to maximize the video quality and balance between delay and loss performance. However, it suffers from delay performance degradations that may occur due to the retransmission mechanism in MPTCP when packet loss occurs. A Distortion aware concurrent multipath transfer scheme, proposed in [1], that aims to reduce the end-to-end video distortion rate by relying on path status estimation, flow rate allocation, and re-transmission control. The available bandwidth on each path is estimated by sending ACK packets one each path. A simple queue is used to characterize packet delay over each component link. In [24], a content aware scheduling technique is proposed to improve the video quality by assessing the video content parameters (e.g., video distortion level, data loss rate, and end to end delay of the frame) in an appropriate route for scheduling the video frame using a decision procedure at server side. A TCP friendly congestion control algorithm is used and packets are distributed to different paths on frame level. In [22], the authors propose energy-distortion mechanism to provide quality-guaranteed video streaming with minimization energy consumption of mobile devices over heterogeneous networks. The minimization of energy consumption was reached though video flow rate allocation algorithm using an analytical energy distortion framework under the target video quality constraint. The result shows that for the same video quality EDAM can with lower energy consumption compared to the energy consumed using the traditional MPTCP. A path failure (PF) detector is proposed in [19] were paths that experience

multiple timeout events are classified as "Potential failed" (PF) and not used for data transmission, while the data will be delivered using the active paths. Concurrent multipath transfer (CMT)-PF reduces the detection latency of link failures and improves CMT's throughput. The current solutions enable the flows or paths to be setup between a client and a single server without exploiting service replication, which is important to prevent server overload issue because of the high demand for high quality video. Besides, transport layer solutions require kernel modifications at both server and client sides, in addition to the limitation that many network operators do not allow MPTCP traffic to pass through their middleboxes. Concerning the current solutions at the application level, they require server side modification without taking the upsides of dynamic adaptive video streaming over HTTP when using the multipath communication. Subsequently, there is a need to build up an efficient video streaming solution from multi-sources to multihomed mobile users implemented over the DASH standard.

III. PROPOSED SOLUTIONS

We develop a selection mechanism between different networks to exploits multihoming feature to maximizing the QoE of the DASH standard clients. Basically, the use of DASH permits the video content to be stored as segments of different qualities known as representations. The DASH player start streaming the content by loading the Media Presentation Description (MPD) which is a manifest file that describe the video content (video and segment length, video qualities and etc.). Then the player start downloads the video segments in sequence over HTTP where an Adaptive Bit rate (ABR) is executed to permit the client to choose different qualities of downloaded segment according to its current network conditions [21]. Thus, the client can watch the video at the highest available quality, and continuously without any freezes. The client has the choice to decrease the bitrate when the network throughput degrades and increase the bitrate when the network throughout improves. In Dash design, there is a buffer at client side permit to store a number of segments then played out when the user requires. Rebuffering occurs when the buffer is empty and not more segments to play out, thus the video playback must freeze until receiving the nest segments. ABR algorithms are classified into three types: throughput, buffer, and hybrid rules. Throughput Rule like PANDA [16] and Festive [13] focus on the available network throughput between the player and the server where the requested quality of next downloaded segment is based on the estimation of current network throughput. Buffer Rule like Bola [20] and BBA [12], use the buffer level as indictor of network throughout. It chooses higher quality when the buffer level is higher and lower when the buffer level is lower. Hybrid Rule like MPC [25] and ELASTIC [8] use both throughput and buffer level to take the advantages of both. A key goal of our model is to provide higher QoE in terms of higher bitrate and fewer rebuffers, to a multimomed mobile dash client taking the advantage of CDN where the video can be replicated at multiple servers. Multi-Armed bandit algorithms are implemented on client DASH protocol, permitting the user

to watch the video in highest quality by switching between different available networks.

A. Model

- t_C: Segment duration.
- D: Video duration.
- C: Number of segments , which is equal to $\lceil \frac{D}{t_C} \rceil$.
- k: Time is divided into steps with a duration of K. The duration of the time step is equal to $c \times t_C$, where c is the fixed number of requested segments to be downloaded in a time step.
- $\mathcal{I} = \{I_i\}_{i=1..N_I}$: Available networks N_I
- $n_i(k)$: Number of times that network i is selected until time step k.
- $d(k)$: The selected network at time step k (i.e. decision).
- $r(k)$: The reward received at time step k, which is the average bitrate measured at network $d(k)$.
- $\mathcal{Q} = \{q_j\}_{j=1..N_Q}$ is the set of available N_Q video bitrates.

B. MAB Algorithm- Epsilon Greedy

Algorithm 1 MAB Algorithm- Epsilon Greedy

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

MAB algorithms [26] aim mainly to select the network which maximize the QoE. we start by the ϵ-greedy algorithm which is one on the main algorithms behind science of decision, and symbolizes the balance of exploration versus exploitation. The algorithm keeps track of the cumulative average reward $\overline{r}_i$ of each network i in $\mathcal{I}$, that is given by:

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

At each time step k, exploitation or explorations may occur.Exploitation consist of choosing the network that has the best average return based on what we know of the circumstances with a probability $1-\epsilon$. exploration consist of choosing a random network with probability ϵ (see lines 6 to 9) using the function the latter case, a discover phase is allowed by selecting a random network, using the $getRandNetwork(\mathcal{I})$.

The ϵ parameter represent the percentage of time/trials dedicated for exploration. It is typical be around 10%, but in our experiments we study the impact of this parameter.For the selected network i, the function $getNSegments(c, q_{init}, i)$ is used to download c segments, q_{init} being the quality of

the first segment. The qualities for subsequent segments will be adaptively determined by the ABR algorithm. The funtion $getNSegments(c, q_{init}, i)$ returns the reward, which is the average bit rate measured at network i, and the average measured quality. In this section, the quality of requested segments is determined according to the ABR rules used by default in DASH as described before (Throughput and Buffer rules).

C. MAB algorithm - Epsilon Greedy with ABR modification

According to abr rules, playing the video at the highest available quality and continuously without any freeze are the main factors that improve the QoE. In the case of multihoming, the presence of multiple networks may cause some false choices of quality due to different network characteristics. Request a segment at a high bitrate, based on the assumption on first network conditions, through a different network that may have lower throughput than the first network, will cause the client to rebuffed because the player is unable to fetch this required segment at the quality requested. In this real scenario, the characteristics of different available networks should be an additional indicator to ensure that ABR is well working and improve the QoE. A simple modification is proposed in Algorithm 2, to solve this problem, consist of setting the quality of the first requested segment on the switched network equal to the last quality measured on the same network.

Algorithm 2 MAB Algorithm- Epsilon Greedy with Abr modification

---
1: **for** $i = 1$ to N_I **do**   ▷ bootstrap
2:     $\overline{r}_i, q_{init}(i) = getNSegments(c, DEFAULT, i)$
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 network
8:     **else**
9:        $d(k) = getRandNetwork(\mathcal{I})$ ▷ random network
10:    $r(k), q_{init}(d(k)) = getNSegments(c, q_{init}(d(k)), d(k))$
11:    $\overline{r}_{d(k)} = \alpha r(k) + (1 - \alpha)\overline{r}_{d(k)}$
12:    $k = k + 1$

In the next paragraph, we study the performance of MAB-Epsilon Greedy without abr modification in different networks scenarios. Then in case the described problem appear, we compare the results to MAB-Epsilon Greedy with abr modification as mention in Algorithm 2.

D. MAB algorithm- Upper Confidence Bounds (UCB) with ABR modifictaion

Upper Confidence Bounds (UCB) is a deterministic algorithm that focuses on exploration and exploitation based on a confidence boundary that the algorithm assigns to each network on each time of exploration. UCB is based on choosing the lever that promises the highest reward under optimistic assumptions (i.e. the variance of your estimate of the expected reward is high, therefore you pull levers that

you don't know that well) [2]. UCB is an allocation strategy achieving logarithmic regret uniformly over n and without any knowing in advance about the reward distributions. It can be briefly said that UCB's main idea is: the principle of optimism in the face of uncertainty in spite of our lack of knowledge in what actions are best. Our optimistic guess will quickly decrease and we'll be urged to switch to a different action (network) in case the guess was wrong. But if we pick well, we'll be able to experience little regret. UCB algorithm is used in multi armed bandit problem where given a set of arms, our goal is to maximize the reward got after using every arm. UCB explores all arms then starts exploiting. The client is multihomed, has multiple interfaces refer to multiple networks or connections. The arms are the networks. These networks can have different throughput's which ensures different rewards. The reward is inferred from the quality of the requested segment. Since the video player does not switch the quality immediately, UCB start by exploring all the networks then try to maximize the quality of the played video. For each available network, an upper-confidence bound is defined that shows the highest guess at the payoff for that arm. With the UCB algorithm, the arm (network) with the highest UCB calculated index is chosen for each transmission. The arm with the highest probability of being "the best arm" is being played. The reward for each network is defined Equation (1) in shows the highest guess at the payoff for that network as well as the number of times the network has been chosen.

$$\overline{r}_i = \overline{r}_i + \sqrt{\left(2\log_2\left(\frac{\frac{C}{c}}{n_i(k)}\right)\right)} \tag{1}$$

As described in Algorithm 3, the bootstrap phase consists of download c chunks at each available network i and calculate the reward of each network. After all the networks have been used, it is the UCB values (reward) turn to determine the network to be selected. After every time step(k), the function $getNSegments(c, q_{init}, i)$ is used to download c segments using the selected network i, q_{init} being the quality of the first segment. Then, the UCB value (reward) is updated for every network. we set the quality of the first requested segment on switched network equal to the last quality detected on the same network. Finally, select the network with the highest UCB and switch to.

Algorithm 3 MAB Algorithm- UCB Algorithm with Abr modification

---
1: **for** $i = 1$ to N_I **do**   ▷ bootstrap
2:     $\overline{r}_i, q_{init}(i) = getNSegments(c, DEFAULT, i)$
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:     $d(k) = I^*$ ▷ select the best network
7:     $n_{d(k)} = +1$
8:     $r(k), q_{init}(d(k)) = getNSegments(c, q_{init}(d(k)), d(k))$
9:     $\overline{r}_{d(k)} = \overline{r}_{d(k)} + \sqrt{(2\log_2(\frac{\frac{C}{c}}{n_i(k)}))}$
10:    $k = k + 1$

IV. ANALYTICAL PART

Moving from theoretical algorithms to applied successful algorithms requires a set of experiments that take into consideration all possibilities. In our proposal, as mention in the Introduction part, we should discover the performance of the proposed algorithms in channel diversity different network conditions between different channels) and intra-channel diversity (network variations inside the same channel). The performance evaluation of our proposed model is achieved through the implementation of an experimental test-bed in order to avoid the dynamicity of the Internet and be able to generate the same network configurations suitable for validating the experimentation scenarios under question. Wondershaper tool is used to limit the throughput of networks, it allows us to set the maximum download rate and/or maximum upload rate of an interface (refer to a network) from the command line.

A. Reference Measurements

we start our experiments by a set of reference measurements.

1) Dash performance in different network speeds: The objective of these measurements is to evaluate how the DASH protocol deal with network speed variations. In order to evaluate our proposed algorithms, we need to compare them to a reference basic scenario presented in Figure 1. In this scenario, a normal DASH client watches a video content of duration 10 minutes stored in a server with 10 different bitrates. In order to evaluate the key of using DASH protocol and standard ABR rule, the same scenario is repeated with different network throughputs controlled by the Wondershaper tool which permit us to set different download/ upload rate on each scenario.

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

As mention before, in order to evaluate the QoE of the player, two main factors are required: the average video rate and re-buffering time. In each scenario, we measure the average video rate which is the video average quality (bitrate) received. Figure 2 presents the results showing that the highest quality could be reached in a fast network throughout and permit us to map the quality to the network throughout.
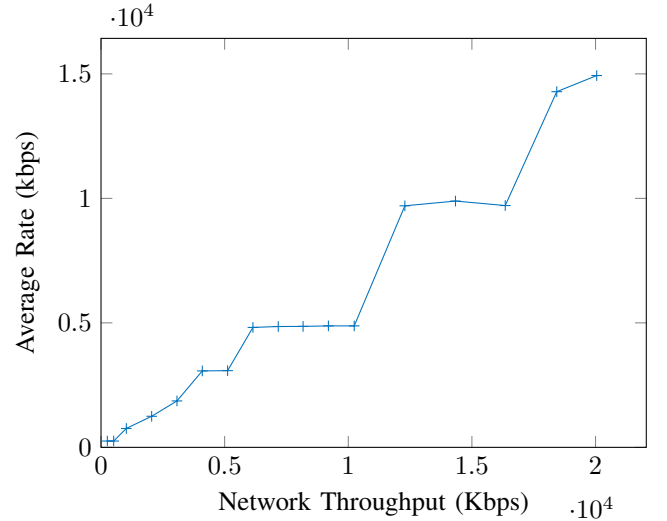

Fig. 2: Video quality variation in function of network speed variation

The second indicator of the QoE is the rebuffering issue, resulting in counting the time of empty buffer occurrences and the duration of each occurrence. Figure 3 shows how many time we notice empty buffer and the duration of each appearance in different network throughout's.
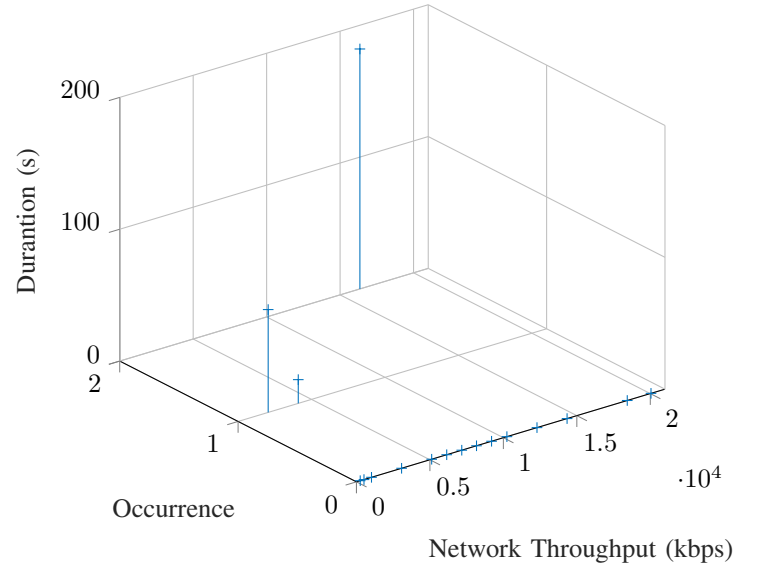

Fig. 3: Duration of empty buffer occurrences as a function of network throughput variation

2) MAB-EPS greedy varitations: In these experiments, our objective is to discover the best value of eps when using the Epsilon Greedy algorithm. In this second experiment set, presented in Figure 4, the client is Mutlithomed DASH player occupied with two network connections and using the same video content but replicated in two mirror server which permit us to cover the concept of CDN network. We implement algorithm 1 at the DASH client side.
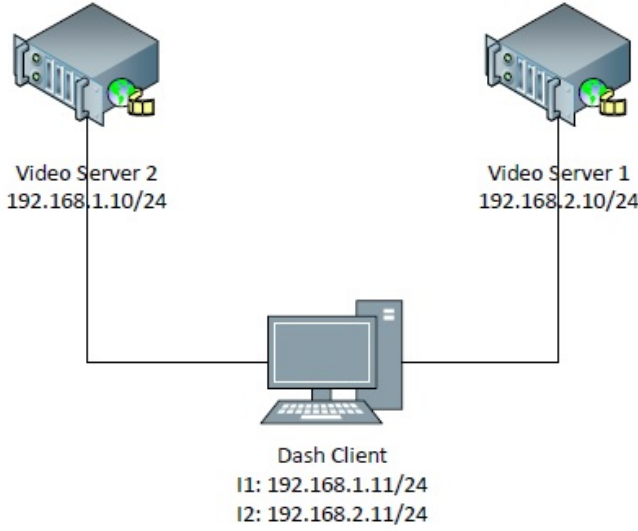
Fig. 4: Experimental environment

Fig. 6: Average Video quality vs Eps value

The Epsilon Greedy algorithm consists of a tradeoff between exploitation and exploration based on the "epsilon" parameter eps. This parameter represents the percentage of time/trials dedicated for exploration, and it is also typical to do random exploration. 81 scenarios were built and run (Figure 5) in order to determine the best value of eps. each scenario consists of watching the video content with different throughput's for network connection two (varies from S_0 to S_i) while fixing the throughput of the first connection to 5120 kbps. Based on the result of the first scenario, we choose 9 different sets of network throughput's based on the value of throughput that affects the average video bitrate. Then the same scenarios were repeated using 9 different valuess of eps.

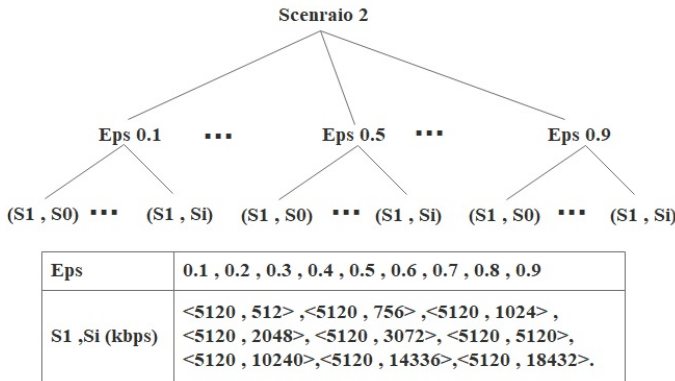| Eps	0.1 , 0.2 , 0.3 , 0.4 , 0.5 , 0.6 , 0.7 , 0.8 , 0.9
S1 ,Si (kbps)	<5120 , 512> ,<5120 , 756> ,<5120 , 1024>, <5120 , 2048>, <5120 , 3072>, <5120 , 5120>, <5120 , 10240>,<5120 , 14336>,<5120 , 18432>.

Fig. 5: Measurement Set 1

In Figure 6, the average video bitrate received is compared between different values of eps. The highest quality is reached when the value of eps is 0.1 which means we have 10% exploration while the quality decreased from 5Mbps to 3Mbps when we have 90% exploration (eps equal to 0.9).
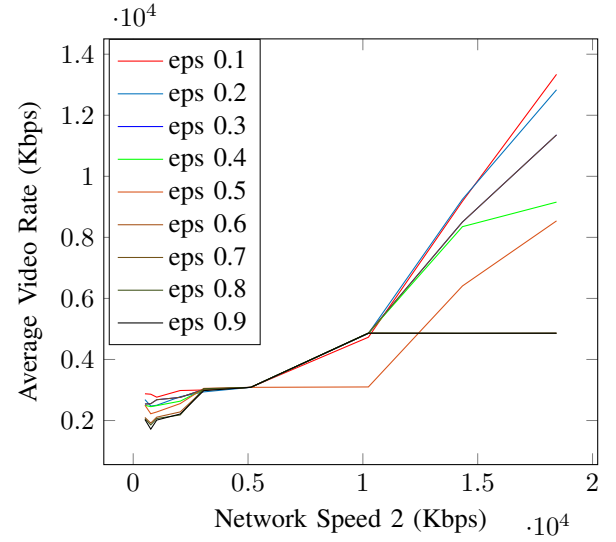
The QoE of the user is considered based on the rebuffering issue as mentioned before. The result of Figure 7 confirms the previous result, of choosing 10% for exploration where 5 seconds freezes appear one time.
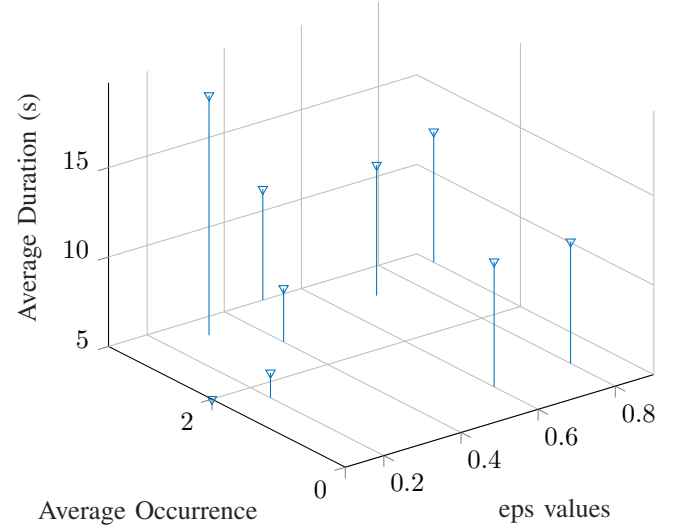

Fig. 7: Average empty buffer occurrence and duration in function of eps variation

According to these results, we set the value of eps to 0.1 in all future experiments for the Epsilon Greedy algorithm.

B. MAB-Epsilon Greedy performance

In order to evaluate the performance of Epsilon Greedy as network selection protocol for a multi-homed Dash client, a new set of measurements were run using 81 different combination values of network throughput's as shown in Figure 8.
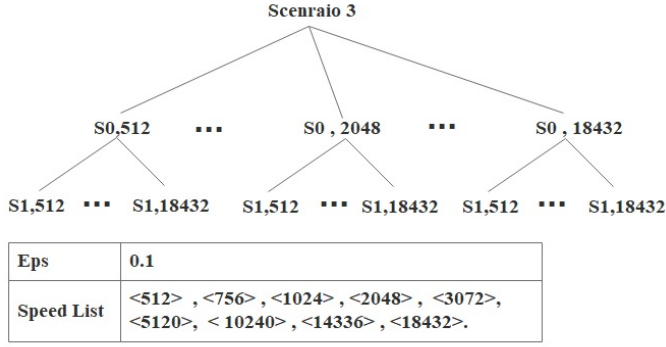
Fig. 8: Measurement Set 2 of Algorithm 1 implementation

For each measurement, we calculate the bitrate enhancement E (Equation 1) by measuring the average bit rate Q_i when using network connection i on each scenario.

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

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 networks alone and to MAB proposed without abr modification. The results show that using multihoming improve the QoE of user expect in 4 scenarios. Different enhancement ranges were observed, up to 80% enhancement in 70% from the measurements and 20% in the remaining measurements.

Fig. 10: Empty Buffer Occurrence and Duration of Algorithm 1

C. DASH-Abr performance when using MAB-Epsilon Greedy

The same measurement set 2 scenarios are repeated after modifying the ABRrule in DASH in order to solve the freeze effected occurred due to different network throughput's between different networks. The results (Figure 11) show the client plays the video continuously without any freezes and the QoE is enhanced up to 85% in 80% of the scenarios. Thus, providing good QoE in the case of multihoming requires some modification in standard ABR rules.

Fig. 11: Average Bitrate Enhancement of Algorithm 2

D. MAB-Epsilon Greedy Vs MAB-UCB performance in intra-networks variation

Based on results in the previous paragraph, we use the same abr modification in Dash in both MAB algorithms (considered before in Algorithm 3). In this section, the same experiment environment presented in Figure 4 is used,we compare the
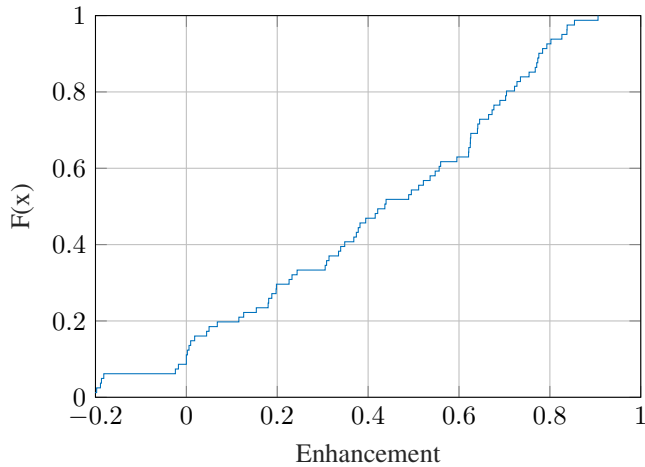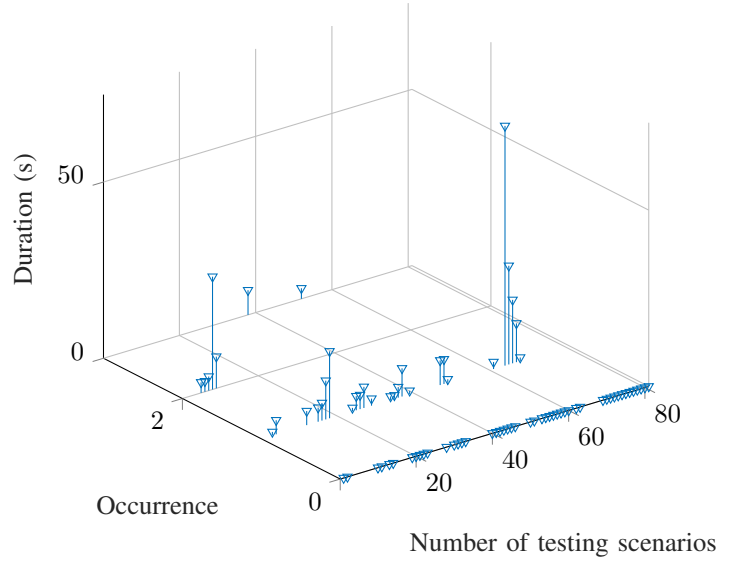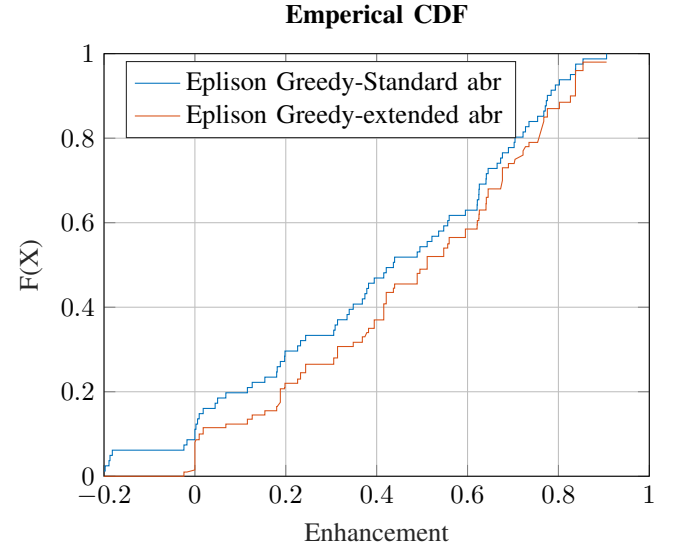

Fig. 9: Average Bitrate Enhancement of Algorithm 1

But the excepted theoretical problem in (III.C) was clearly observed in Figure 10 where the client has to freeze 3 seconds in order to play the video continuously in 41% of the scenarios.

performance of Epsilon Greedy algorithm to UCB algorithm, while the throughput of network 1 is fixed to 5120 kbps and running different scenarios with different speeds of the second connection. The results of Figure 12 show better quality is attained when applying the UCB algorithm especially when there is a big gap between both server's bitrates. For example, in the case where Server 1 bandwidth is 5 Mbps and Server 2 bandwidth is 512 Kbps, the average quality obtained is double with UCB used. We notice that when using the UCB method, the average quality doesn't vary much even when Server 2 bandwidth differs from 512 Kbps to 5 Mbps. This result is anticipated since the UCB chooses the most network speed available that has the highest reward.
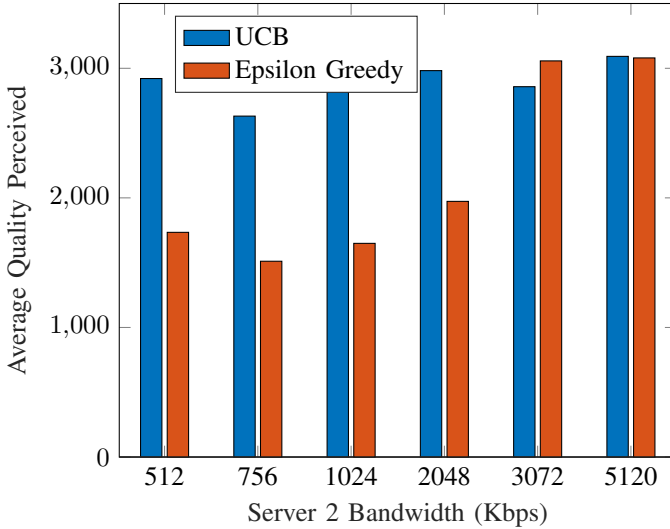

Fig. 13: Empty Buffer Occurrence and Duration of UCB and Epsilon Greedy

E. MAB-Epsilon Greedy Vs MAB-UCB performance in inter-network variations

In these experiments, our objective is to ensure the reliability and effectiveness of the performance evaluated between Epsilon Greedy and UCB algorithm during different inter-network variations, which permit us to ensure and validate the results. Each scenario consists of watching the video file with variable network speeds for network connection two (varies on different times as showing in Table 1), while fixing the speed of the first connection to 5 Mbps. Each scenario is repeated using MAB-Eplison Greedy and UCB algorithm.

Fig. 12: Average quality perceived using UCB and Epsilon Greedy

TABLE I: Network Variation Scenarios

Scenario	Speed 2	Time(min)
1	5 Mbps $\triangleright 10Mbps$	5
2	5 Mbps $\triangleright 18Mbps$	5
3	5 Mbps $\triangleright 1Mbps$	5
4	5 Mbps $\triangleright 512Kbps$	5
5	5 Mbps $\triangleright 1Mbps \triangleright 4Mbps \triangleright 8Mbps$	2
6	5 Mbps $\triangleright 8Mbps \triangleright 2Mbps \triangleright 4Mbps$	2
7	5 Mbps $\triangleright 8Mbps \triangleright 10Mbps \triangleright 4Mbps$	2
8	5 Mbps $\triangleright 2Mbps \triangleright 1Mbps \triangleright 4Kbps$	2

Furthermore, the video experience is affected by the empty buffer occurrence and the duration of the buffer's duration. Figure 13 show that UCB has lesser number of occurrence than Epsilon Greedy which can be interpreted with a small delay and rebuffering. Adding to that, the buffer's duration in UCB is minor compared to Epsilon Greedy which incurs fewer frequent freezing and annoying interruptions. The QoE of the user is also determined based on the count of freezes occurrence with different periods. Figure 13 shows that a maximum re-buffering duration of 17 seconds in UCB appears to be more desirable than the 66 seconds in Epsilon Greedy.
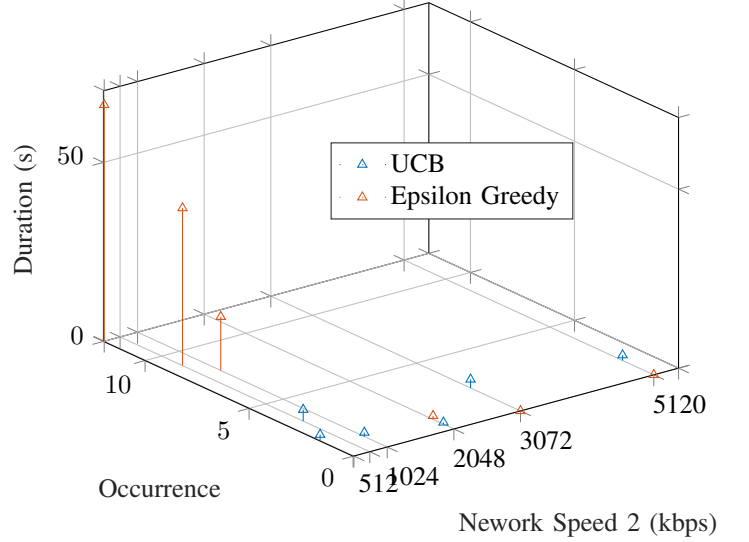
As illustrated in Figure 14, the average qualities perceived in approximately all scenarios were closed to each other in all cases expect in scenario 2. In scenario 2, the average bitrate in UCB was double the average bitrate in Epsilon Greedy. This scenario is when the server 2 speed has increased twice the server 1 speed. In our case, higher bitrate could be achieved when using UCB. Furthermore, as we notice, the average quality obtained in Experiment 1 is much greater than the one in Experiment 2 which assures that network variations affect negatively on the overall output and performance for the Dash client and degrade his QoE.
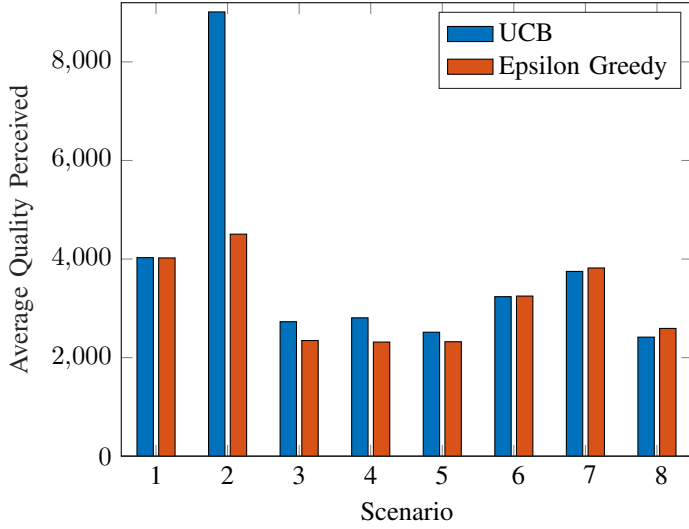
Fig. 14: Average quality perceived using UCB and Epsilon Greedy for testing scenarios

As visualized in Figure 15 , the number of occurrences in Epsilon Greedy is superior to the one in UCB. We can also tackle the issue that especially in scenarios 3 and 4, where server 2 speed decreases. The duration gets higher in both cases, but UCB is much higher than Epsilon Greedy especially for the set of scenarios (5,6,7,8) when more fluctuations in network Speed appear. Therefore, Epsilon Greedy's performance is much annoying to the user when the network flutters.
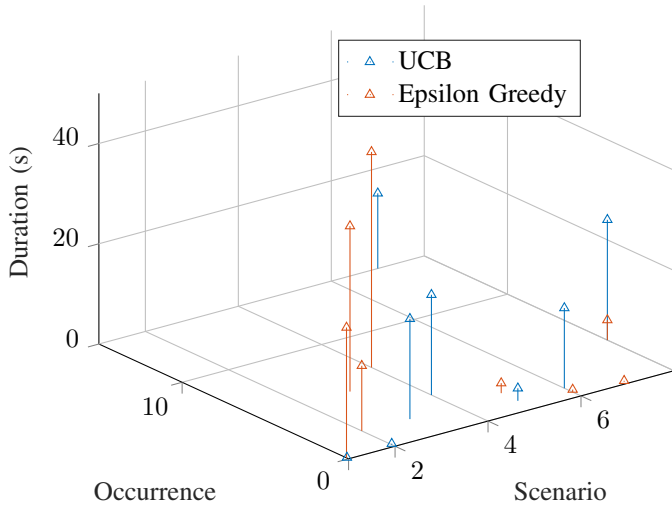

Fig. 15: Empty Buffer Occurrence and Duration of UCB and Epsilon Greedy for testing scenarios

V. CONCLUSIONS

This work presents two MAB algorithms for multihomed DASH clients for the dynamic selection of network connection aiming for a better adaptive video streaming. Our experiments results show that heterogeneous and variable network conditions lead to different performance for both MAB algorithms.

For different speeds between different connections but stable network conditions on each connection,the results show that UCB exceeds Epsilon Greedy in terms of goodput perceived by the Dash client but empty buffer occurrences still exist in both situations.The ϵ-greedy algorithm takes the best action most of the time, but does random exploration occasionally while UCB ensures we select optimal actions as $t \to \infty$ marking smart exploration. Thus, the UCB algorithm is able to find the balance between exploring unfamiliar options, and exploiting the winning variant. But for unstable network conditions on each connection, the results show that ϵ-greedy is more stable than UCB. Our future work will focus how use the availability of multiple connection to introduce the parallel streaming over DASH, which will be a new case of Multihoming to enhance the video quality perceived.

REFERENCES

[1] S. Agarwal, S. De, and J. B. Seo. Cognitive multihoming: Maximizing network utility over cr-assisted cellular network. In *2015 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, 2015.

[2] S. Boldrini, J. Fiorina, and M. Di Benedetto. Introducing strategic measure actions in multi-armed bandits. In *2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops)*, pages 41–45, 2013.

[3] D. H. Bui, K. Lee, S. Oh, I. Shin, H. Shin, H. Woo, and D. Ban. Greenbag: Energy-efficient bandwidth aggregation for real-time streaming in heterogeneous mobile wireless networks. pages 57–67, Dec 2013.

[4] Y. Cao, D. Yu, L. Zeng, Q. Liu, F. Wu, X. Gui, and M. Huang. Towards efficient parallel multipathing: A receiver-centric cross-layer solution to aid multipath tcp. In *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 790–797, 2019.

[5] Y. Cao, L. Zeng, Q. Liu, G. Lei, M. Huang, and H. Wang. Receiver-assisted partial-reliable multimedia multipathing over multi-homed wireless networks. *IEEE Access*, 7:177675–177689, 2019.

[6] Y. Chen, D. Towsley, and R. Khalili. Msplayer: Multi-source and multi-path video streaming. *IEEE Journal on Selected Areas in Communications*, 34(8):2198–2206, Aug 2016.

[7] Cisco. Cisco visual networking index: Forecast and trends, 2019. Available at https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

[8] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo. Elastic: A client-side controller for dynamic adaptive streaming over http (dash). In *2013 20th International Packet Video Workshop*, pages 1–8, 2013.

[9] S. Ferlin, S. Kucera, H. Claussen, and Ö. Alay. Mptcp meets fec: Supporting latency-sensitive applications over heterogeneous networks. *IEEE/ACM Transactions on Networking*, 26(5):2005–2018, 2018.

[10] Y. Go, H. Noh, G. Park, and H. Song. Hybrid tcp/udp-based enhanced http adaptive streaming system with multi-homed mobile terminal. *IEEE Transactions on Vehicular Technology*, 68(5):5114–5128, 2019.

[11] J. Huang and Z. Lin. Notice of violation of ieee publication principles: Group-aware delay-constrained video transmission over multihomed device-to-device networks. *IEEE Access*, 5:2651–2664, 2017.

[12] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation. *ACM SIGCOMM Computer Communication Review*, 44:187–198, 02 2014.

[13] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking*, 22(1):326–340, 2014.

[14] Jiyan Wu, Jingqi Yang, and Junliang Chen. Loss tolerant bandwidth aggregation for multihomed video streaming over heterogeneous wireless networks. pages 2956–2962, Dec 2013.

[15] S. Lee, H. Roh, and N. Lee. Enhanced quality adaptation scheme for improving qoe of mpeg dash. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 357–362, 2017.

[16] 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 Journal on Selected Areas in Communications*, 32(4):719–733, 2014.

[17] M. Ma, Z. Wang, K. Su, and L. Sun. Understanding the power of smartrouter-based peer cdn for video streaming. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, 2016.

[18] C. Mueller, S. Lederer, C. Timmerer, and H. Hellwagner. Dynamic adaptive streaming over http/2.0. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2013.

[19] Preethi Natarajan, Nasif Ekiz, Paul D. Amer, and Randall Stewart. Concurrent multipath transfer during path failure. *Computer Communications*, 32(15):1577 – 1587, 2009.

[20] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. Bola: Near-optimal bitrate adaptation for online videos. *IEEE/ACM Transactions on Networking*, 28(4):1698–1711, 2020.

[21] J. Vlaović, S. Rimac-Drlje, F. Vranješ, and R. P. Kovač. Evaluation of adaptive bitrate selection algorithms for mpeg dash. In *2019 International Symposium ELMAR*, pages 73–76, 2019.

[22] J. Wu, B. Cheng, M. Wang, and J. Chen. Quality-aware energy optimization in wireless video communication with multipath tcp. *IEEE/ACM Transactions on Networking*, 25(5):2701–2718, Oct 2017.

[23] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen. Streaming high-quality mobile video with multipath tcp in heterogeneous wireless networks. *IEEE Transactions on Mobile Computing*, 15(9):2345–2361, Sep. 2016.

[24] J. Wu, C. Yuen, M. Wang, and J. Chen. Content-aware concurrent multipath transfer for high-definition video streaming over heterogeneous wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 27(3):710–723, March 2016.

[25] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. *ACM SIGCOMM Computer Communication Review*, 45:325–338, 08 2015.

[26] J. Zhang, J. Tang, and F. Wang. Cooperative relay selection for load balancing with mobility in hierarchical wsns: A multi-armed bandit approach. *IEEE Access*, 8:18110–18122, 2020.