Open Access
2021, 18(4) Supplement: 1441-1447

Baghdad Science Journal

P-ISSN: 2078-8665
E-ISSN: 2411-7986

DOI: https://dx.doi.org/10.21123/bsj.2021.18.4(Suppl.).1441

An Experimental Study of the Server-based Unfairness Solutions for the Cross-Protocol Scenario of Adaptive Streaming over HTTP/3 and HTTP/2

*Chanh Minh Tran** *Tho Nguyen Duc* *Phan Xuan Tan* *Eiji Kamioka*

School of Engineering and Science, Shibaura Institute of Technology, Japan.
*Corresponding author: nb20502@shibaura-it.ac.jp
E-mails: nb20501@shibaura-it.ac.jp , tanpx@shibaura-it.ac.jp , kamioka@shibaura-it.ac.jp

Received 15/10/2021, Accepted 14/11/2021, Published 20/12/2021

Abstract:

Since the introduction of the HTTP/3, research has focused on evaluating its influences on the existing adaptive streaming over HTTP (HAS). Among these research, due to irrelevant transport protocols, the cross-protocol unfairness between the HAS over HTTP/3 (HAS/3) and HAS over HTTP/2 (HAS/2) has caught considerable attention. It has been found that the HAS/3 clients tend to request higher bitrates than the HAS/2 clients because the transport QUIC obtains higher bandwidth for its HAS/3 clients than the TCP for its HAS/2 clients. As the problem originates from the transport layer, it is likely that the server-based unfairness solutions can help the clients overcome such a problem. Therefore, in this paper, an experimental study of the server-based unfairness solutions for the cross-protocol scenario of the HAS/3 and HAS/2 is conducted. The results show that, while the bitrate guidance solution fails to help the clients achieve fairness, the bandwidth allocation solution provides superior performance.

Keywords: Adaptive Streaming, Cross-Protocol, HTTP/3, QUIC, Server, Unfairness.

Introduction:

Although official standardization is not yet available, the HTTP/3 [1] has become a hot research topic in recent years, especially for improving the adaptive streaming over HTTP (HAS). The HTTP/3 differs from its successors HTTP/2 and HTTP/1.1 by replacing the traditional TCP protocol with the novel QUIC protocol [2] in the transport layer. The QUIC protocol is actually based on UDP with enhanced TCP-like features implemented on the user space that promises to solve TCP Head-of-Line blocking, fasten connection handshakes, improve multiplexing and congestion control, etc. Such enhancements are expected to reduce network delay and improve bandwidth utilization, thus benefiting the delay- and/or time-sensitive Internet applications like HAS. For this reason, since the introduction of the HTTP/3 and QUIC, a considerable amount of research has focused on investigating their influences on the performance of existing HAS implementations.

In HAS, a video is encoded in multiple quality versions (represented by the video bitrates) and each version is chunked into fixed-duration segments. Then, when streaming the video, the adaptive bitrate selection algorithm (ABR) at a HAS client continuously estimates the available bandwidth in order to decides the best suitable video bitrate [3]. As bandwidth is the most important measurement for bitrate adaptation, it is obvious that the performance of a HAS client can be improved if the bandwidth is optimally utilized. As bandwidth utilization is among the main improvement goals of the HTTP/3 and its transport QUIC, various research has investigated and provided interesting assessments on whether the adaptive streaming over HTTP/3 (HAS/3) could actually outperform one over HTTP/2 (HAS/2) and/or over HTTP/1.1 (HAS/1.1) [4, 5]. However, similar works for the multiclient scenario, where the bandwidth competition occurred, were surprisingly limited.

It has been found in existing HAS/1.1 and HAS/2 studies that, when multiple video clients compete for a bottleneck bandwidth, the unfairness in bitrate selection among them is likely to happen, where some clients may request higher bitrates than the others sharing the same network [6]. As proven in these studies, such a problem is due to the mismatch of the so-called ON-OFF periods among the clients, which describe the downloading-idle states of the

ABR on the application layer. Based on this finding, existing research on solving the unfairness problem mainly focused on proposing advanced ABRs that were more fairness-aware [7, 8]. On the other hand, there also have been a few attempts to investigate such a problem of the HAS/3. Specifically, due to the potential unavailability of the HTTP/3 due to the UDP nature of its transport QUIC, many works have focused on the cross-protocol scenario where HAS/3 clients compete with HAS/2 and/or HAS/1.1 clients [9, 10, 11]. Nevertheless, those works only provide observations of the general non-fairness-aware ABRs.

In our recent study [12], it has been proven that, in a cross-protocol scenario between HAS/3 and HAS/2 (and/or HAS/1.1), the HAS/3 clients always experience higher bitrates than the HAS/2 and/or HAS/1.1 clients. Such a behavior is because of the difference in the congestion control mechanism between the QUIC transport of HTTP/3 and the TCP transport of the HTTP/2 and HTTP/1.1 [13]. That is, the QUIC protocol helps the HAS/3 clients gain more congestion window, thus consuming higher bandwidth than the competing TCP-based clients. As a result, the application layer fairness ABRs cannot efficiently solve such a cross-protocol unfairness. Additionally, the work also hints at the possibility of applying a server-based fairness solution as it can monitor and manipulate the transport layer better than client-based ABRs. Thus, it is important to investigate whether the server-based can actually improve the fairness of the cross-protocol HAS flows.

In this manner, this paper presents an experimental investigation of the fairness solutions for the HAS/3, with regard to the cross-protocol scenario (i.e., HAS/3 client versus HAS/2 client). It should be noted that, due to diminished usage and features, this work does not consider the HAS/1.1 for evaluation. Based on the experiment results, it is reported that, for the cross-protocol scenario, not all server-based solutions provide optimal performances. Particularly, it has been tested that the bitrate guidance approach fails to improve the fairness of the cross-protocol clients as it doesn’t eliminate the bandwidth competition among them. Whereas, the bandwidth allocation approach provides superior performance as each client is assigned a separate bandwidth slice beforehand.

In the remainder of this paper, firstly, a brief overview of the related works is presented. After that, the experimental setup is described. Then, the experimental results are analyzed and discussed in detail. Finally, the conclusion of this work is provided.

Related Works:

The QUIC transport protocol of the HTTP/3 actually runs on top of the UDP, which has been exposed to several security risks [14] As a result, there are realistic scenarios that some HAS clients may fail to use the HTTP/3 as the UDP is blocked, and have to fall back to HTTP/2 or HTTP/1.1 instead. For this reason, streaming providers should consider having their services available in both HTTP/3 and the former versions. This raises the importance of investigations about the cross-protocol performance among the QUIC-based HAS/3 and the TCP-based HAS/2 and HAS/1.1 clients.

In [9], a performance study of some existing general ABRs with the HTTP-over-QUIC client, which is the former name of the HTTP/3, showed that such a kind of client performed worse than the HTTP-over-TCP client in terms of video bitrates. Contrarily, the work in [10] and [11] concluded that the HAS over QUIC clients always tended to unfairly experience higher bitrates than the competing HAS over TCP. It should be noted that, these studies are actually outdated as the employed a deprecated implementation of the HTTP/3 and QUIC. Meanwhile, utilizing the latest documentation and implementation of the HTTP/3 and QUIC, our recent work [12] indicated that the HAS/3 clients could obtain higher bandwidth than the HAS/2 clients due to the enhancements in the transport QUIC. As a result, the client-side fairness ABR was proven ineffective to balance the bitrate selection between the HAS/3 and HAS/2 clients since the problem originated from the transport layer.

However, in HAS/2 and HAS/1.1, a few works have investigated the server-based solutions for solving the unfairness. In [15], a bitrate guidance scheme was proposed that allowed the server to suggest the maximum available bitrate the clients could request. While in [16], a bandwidth allocation module was deployed that calculated the fair portion of the bottleneck bandwidth and assign it separately to each client. For these types of solution, the fairness control decisions were made from the server, which was supposed to have a broader view and manipulation of the transport layer than the HAS clients. For this reason, it is highly possible that such server-based solutions also show promising results for solving the cross-protocol unfairness between HAS/3 and HAS/2 clients. In the following section, our methodology and experimental settings for clarifying such an expectation are presented.

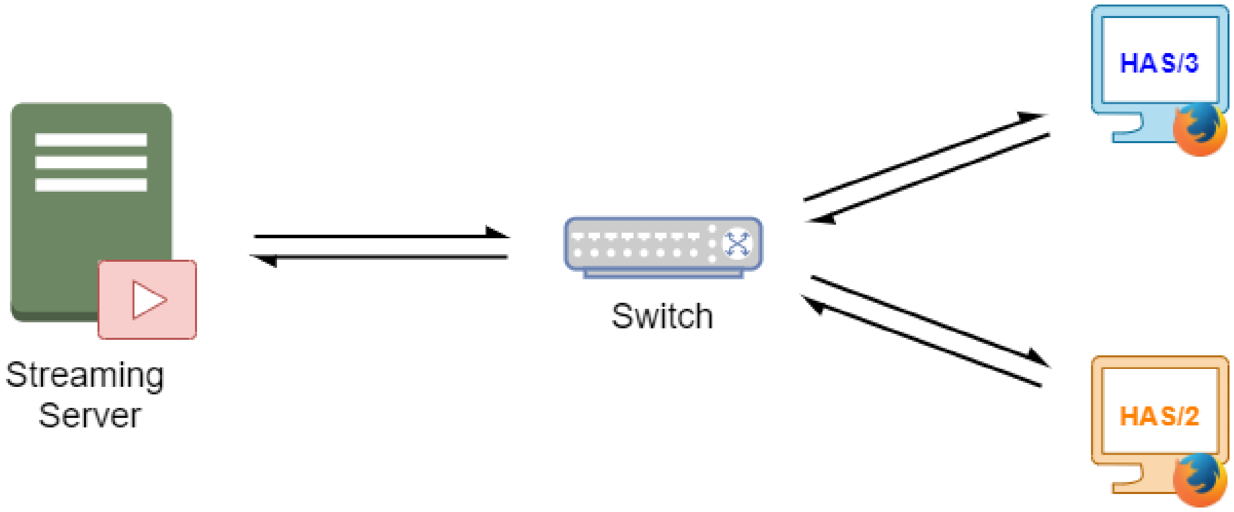

Figure 1. The experiment topology.

Experimental Setup:

Figure 1 depicts the topology used in this experiment. The experimental settings were similar to that of [12]. The server served the HTTP/3 based on *quic-go* v0.20.1, which supported the drafted version 34 of the HTTP/3 and QUIC – the latest version at the time this conducted was experiment. For the HTTP/2, the server simply used the native *http* package of Golang. For the HAS service, the *dash.js* framework was deployed as a web application on the server, which was run via the Firefox web browser at the client machines. In order to emulate the cross-protocol competition, 1 HAS/3 client and 1 HAS/2 client were deployed that competed under a shared 3 mbps bandwidth. Both the server and the clients were the virtual machines running 4-core Ubuntu 20.04 LTS with 4GB of RAM and were actually virtualized with a physical Core i5 machine running Ubuntu 20.04 LTS with 80GB of RAM.

In this experiment, the server stored 300 2-second segments of the open-source Big Buck Bunny video in 11 bitrates, i.e. {100, 200, 300, 500, 700, 900, 1200, 1500, 2000, 2500, 3000} (kbps). The maximum buffer of the client's player was set to 30 seconds. At the client side, as similar to [12], the performance of two ABRs was tested, namely the *dash.js* and the *FESTIVE* [7]. The dash.js was a general ABR based on its own throughput-based rules and the BOLA rules [17], which were not utilized for tackling the unfairness problem. Meanwhile, the FESTIVE was among the most famous baseline for solving the unfairness from the client-side. It employed the harmonic mean measurement of bandwidth, gradual and stateful quality updates and randomized segment download scheduler.

The performance of both of these ABRs were investigated in combination with the following server-based solutions:

- ***Normal***: No server-based solution was deployed. The clients naturally selected the bitrate based on theirs ABRs.
- ***Bitrate Guidance***: The server informed the clients the highest possible bitrate they could request so that the fair share of the bandwidth would not be exceeded. This was similar to the works in [15].
- ***Bandwidth Allocation***: The server allocated a fair bandwidth slice explicitly for every client. This aligned with the solutions in [16].

In order to assess the performance of the evaluated solutions, the following metrics were considered:

- ***Unfairness index***: The unfairness index was utilized as a numerical measurement for the unfairness condition among the clients. It was calculated based on the Jain Fairness index [18], whose detail can be found in [12]. A smaller unfairness index determined a fairer bitrate selection.
- ***Average bitrate***: The average bitrate of a client throughout its streaming session was calculated to find out which type of client requested higher bitrates than the other.
- ***Moving average congestion window***: As found in our previous work [12], the HAS/3 clients were able to obtain higher congestion window than the HAS/2 clients, leading to unfair bandwidth utilization. Therefore, in this experiment, the moving average congestion window was measured to judge whether such a behavior still occurred.

Each server-based solution described above was run 10 times with each of the client-side ABR. In the following section, the average results of the performance metrics are provided and discussed.

Results and Discussion:

Table 1 and 2 summarizes the unfairness index and average bitrate, respectively, of all server-based solutions and client-side ABRs.

It can be inferred from the Table 1 that, when utilizing the FESTIVE, the unfairness was reduced compared to the dash.js, regardless of the server-based solutions. This showed the efficiency of the FESTIVE in solving the unfairness of the HAS. On the other hand, while the performances of the normal and bitrate guidance solution with both ABRs were relatively comparable, the bandwidth allocation solution showed drastic improvements that nearly reached the minimum value of the unfairness index. Taking a look at the Table 2, it can be observed that the average bitrates of the HAS/3 clients running both ABRs under the normal server were always noticeably higher than the HAS/2 clients. Such a performance was expected as confirmed in our previous work [12]. Surprisingly, the bitrate guidance server also produced a relatively similar performance. Meanwhile, all clients under the bandwidth allocation server provided almost identical average bitrates. This indicated that the bitrate guidance solution could not help the HAS/3 and HAS/2 clients select fairer bitrates when they competed, while the bandwidth allocation solution succeeded.

In other to investigate such performances, Fig. 2, 3 and 4 illustrate the time-varying bitrate selection and moving average congestion window of the normal, bitrate guidance and bandwidth allocation solution, respectively. Due to similar behavior, only one representative run is shown for each solution and ABR.

Table 1. The unfairness index of the evaluated server-based solutions and client-side ABRs.

	Normal	Bitrate Guidance	Bandwidth Allocation
dash.js	0.1498	0.1323	0.0086
FESTIVE	0.0840	0.0885	0.0006

Table 2. The average bitrate of the evaluated server-based solutions and client-side ABRs.

	Normal		Bitrate Guidance		Bandwidth Allocation	
	HAS/3	**HAS/2**	**HAS/3**	**HAS/2**	**HAS/3**	**HAS/2**
dash.js	1410	1144	1365	1090	1153	1172
FESTIVE	1156	954	1176	964	1031	1032

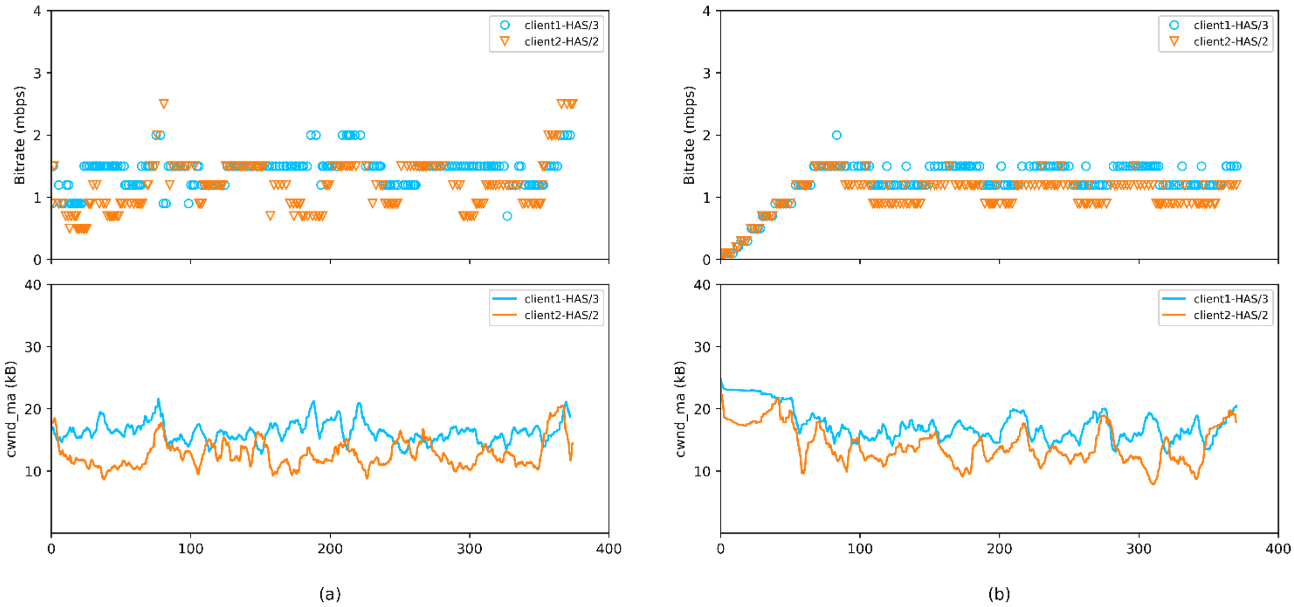

Figure 2. The time-varying bitrate and moving average congestion window of the normal server with a) the dash.js, and b) the FESTIVE.

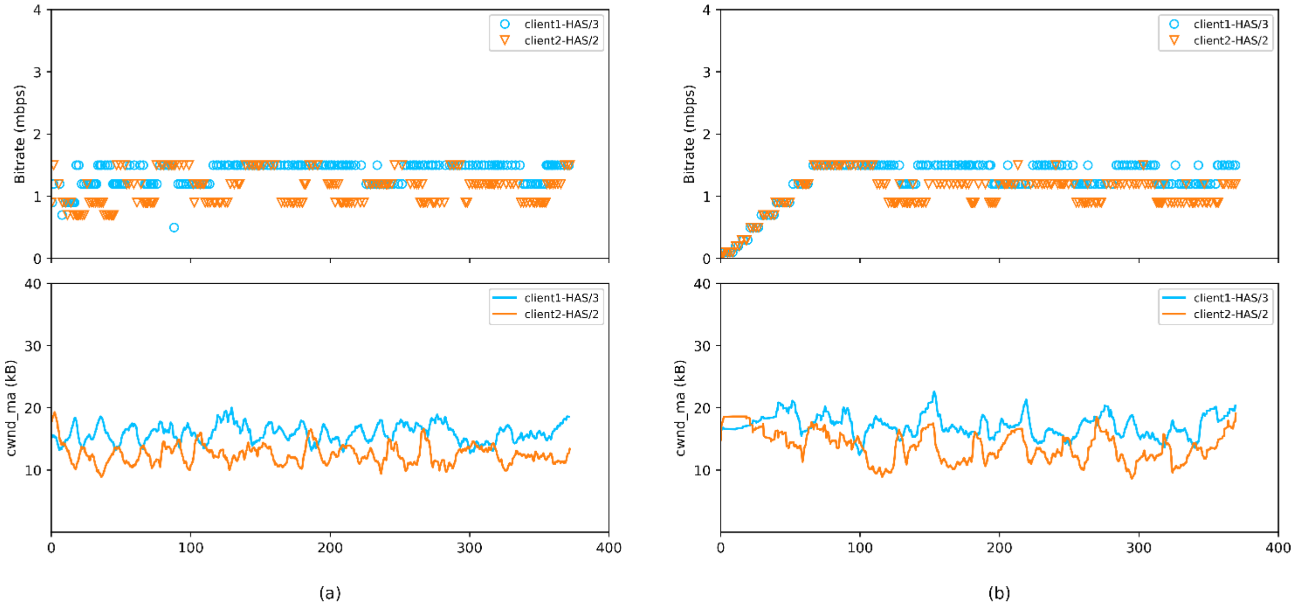

Figure 3. The time-varying bitrate and moving average congestion window of the bitrate guidance server with a) the dash.js, and b) the FESTIVE.

From Fig. 2, it can be seen that the congestion window of the HAS/3 clients tended to be higher than the HAS/2 clients with both ABRs. As a result, the bitrates of the HAS/3 clients were higher for most of the time. Moreover, the bitrate selections of both clients occasionally exceeded the fair bandwidth (i.e., 1.5 mbps as the maximum bandwidth was set to 3 mbps). While in Fig. 3, the bitrates of the clients were controlled not to exceed such a fair share as they had been informed by the server beforehand. Nevertheless, it was shown that the behaviors of the congestion window were similar to those of the normal server: the HAS/3 still obtained higher congestion window. Therefore, although the maximum bitrate was limited, the HAS/3 clients were still able to request higher bitrate as it consumed a larger share of the bandwidth. Such a performance indicated that the bitrate guidance solution could not effectively solve the cross-protocol unfairness between HAS/3 and HAS/2 clients as it could not balance their bandwidth usage.

On the other hand, inferring from Fig. 3, the bandwidth allocation solution significantly outperformed the other solutions by providing almost perfectly fair bitrate selection for the clients. This could be explained that, because each client received and utilized its own slice of bandwidth, it basically did not compete with one another and freely maximized its bitrate. This also explained why their moving average congestion window lines were more stable than the other solutions. Then, as the bandwidth was assigned equally, both clients ended up selecting similar bitrates. It is also

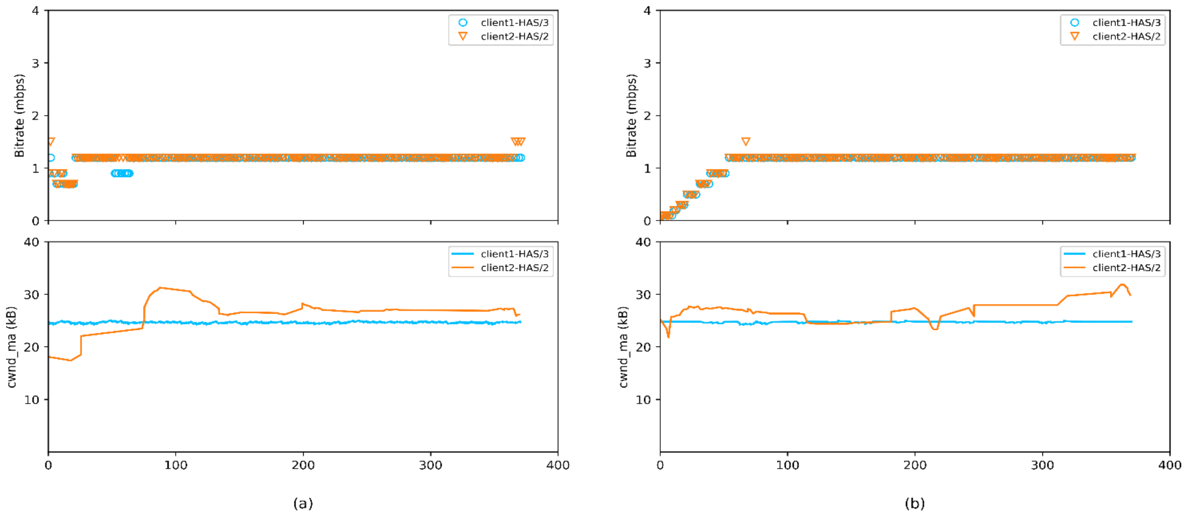

Figure 4. The time-varying bitrate and moving average congestion window of the bandwidth allocation server with a) the dash.js, and b) the FESTIVE.

interesting to note that, when utilizing the bandwidth without competition, the HAS/2 actually increased the congestion window higher than the HAS/3, which was why the average bitrates of the former were slightly higher than the latter. Nevertheless, such a difference was insignificant and it was fair to conclude that the bandwidth allocation succeeded in reducing the unfairness of the cross-protocol scenario of the HAS/3 and HAS/2 clients.

In summary, it was found that the bitrate guidance solution failed to ensure the fairness of the bitrate selections between the HAS/3 and HAS/2 clients because it couldn't balance their bandwidth consumption. On contrary, since the bandwidth competition was eliminated, the bandwidth allocation solution provided a superior performance that successfully helped the clients request bitrates fairly towards each other. Despite such outperformance, the server-based approach has been questioned about its real-life feasibility, in terms of complexity and scalability [19, 20]. Thus, follow-up research should investigate such an approach for the cross-protocol unfairness between the HAS/3 and HAS/2 in a more real-life testbed in order to assess a proper cost-benefit analysis.

Conclusion:

In this paper, an experimental study on the server-based unfairness solutions for the cross-protocol scenario of the HAS/3 and HAS/2 was conducted. Based on the experimental results, it was concluded that the bitrate guidance solution was not an appropriate choice for solving the cross-protocol unfairness problem due to its failure in balancing the bandwidth consumption. On the other hand, it was proven that the bandwidth allocation solution superiorly overcomes such a problem as it could eliminate the bandwidth competition. Based on this finding, for future work, the cost-benefit of applying the bandwidth allocation solution will be investigated in order to deploy such a solution in real life.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- The author has signed an animal welfare statement.
- Ethical Clearance: The project was approved by the local ethical committee in Shibaura Institute of Technology.

Author's contributions:

C.M.T. – conception, design, drafting the MS. C.M.T. and T.N.D. acquisition of data, analysis. P.X.T. and E.K. revision and proofreading. All authors discussed the results and contributed to the final manuscript.

References:

1. Hypertext Transfer Protocol Version 3 (HTTP/3) - draft-ietf-quic-http-34 [Internet]. 2021 [cited 2021 Jun 14]. Available from: https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34
2. QUIC: A UDP-Based Multiplexed and Secure Transport - draft-ietf-quic-transport-34 [Internet]. 2021 [cited 2021 Jun 14]. Available from: https://datatracker.ietf.org/doc/html/draft-ietf-quic-transport-34
3. Hassan et al. PWRR Algorithm for Video Streaming Process Using Fog Computing. Baghdad Sci. J. 2019 Sep;16(3):0667.
4. Seufert M, Schatz R, Wehner N, Gardlo B, Casas P. Is QUIC becoming the New TCP? On the Potential Impact of a New Protocol on Networked Multimedia QoE. In: 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX). Berlin, Germany: IEEE; 2019. p. 1–6.
5. Mondal A, Chakraborty S. Does QUIC Suit Well With Modern Adaptive Bitrate Streaming Techniques? IEEE Netw. Lett. 2020;2(2):85–9.
6. Tran CM, Nguyen Duc T, Tan PX, Kamioka E. FAURAS: A Proxy-Based Framework for Ensuring the Fairness of Adaptive Video Streaming over HTTP/2 Server Push. Appl. Sci. 2020;10(7):2485.
7. Jiang J, Sekar V, Zhang H. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive. IEEE/ACM Trans. Netw. 2014;22(1):326–40.
8. Li Z, Zhu X, Gahm J, Pan R, Hu H, Begen AC, et al. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. IEEE J. Sel. Areas Commun. 2014;32(4):719–33.
9. Bhat D, Rizk A, Zink M. Not so QUIC: A Performance Study of DASH over QUIC. In: NOSSDAV'17. San Jose, CA, USA: Association for Computing Machinery; 2017. p. 13–18.
10. Bhat D, Deshmukh R, Zink M. Improving QoE of ABR Streaming Sessions through QUIC Retransmissions. In: The 26th ACM International Conference on Multimedia. San Jose, CA, USA: ACM; 2018. p. 1616–1624.
11. Arisu S, Yildiz E, Begen AC. Game of Protocols: Is QUIC Ready for Prime Time Streaming? Int J Netw Manag. 2020;30(3):18.
12. Tran CM, Nguyen Duc T, Tan PX, Kamioka E. Cross-Protocol Unfairness between Adaptive Streaming Clients over HTTP/3 and HTTP/2: A Root-Cause Analysis. Electronics. 2021;10(15):1755.
13. QUIC Loss Detection and Congestion Control - draft-ietf-quic-transport-34 [Internet]. 2021 [cited 2021 Jun

14]. Available from: https://datatracker.ietf.org/doc/html/draft-ietf-quic-recovery-34
14. Wangen G, Shalaginov A, Hallstensen C. Cyber Security Risk Assessment of a DDoS Attack. In: Information Security. Springer International Publishing; 2016.
15. Altamimi S, Shirmohammadi S. QoE-Fair DASH Video Streaming Using Server-side Reinforcement Learning. ACM Trans Multimedia Comput Commun Appl. 2020;16(2s):68.
16. Guguen CT, Bolzer FL, Houdaille R. Improving User Experience when HTTP Adaptive Streaming Clients Compete for Bandwidth. SMPTE Motion Imaging J. 2017;126(1):28–34.
17. Spiteri K, Urgaonkar R, Sitaraman RK. BOLA: Near-Optimal Bitrate Adaptation for Online Videos. IEEE/ACM Trans. Netw. 2020;28(4):1698–711.
18. Jain R, Chiu D-M, Hawe W. A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. CoRR [Internet]. 1998;cs.NI/9809099. Available from: https://arxiv.org/abs/cs/9809099
19. Bentaleb A, Taani B, Begen AC, Timmerer C, Zimmermann R. A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. IEEE Commun. Surveys Tuts. 2019;21(1):562–85.
20. Dar et al. Fog Computing Resource Optimization: A Review on Current Scenarios and Resource Management. Baghdad Sci. J. 2019 Jun;16(2):0419.

دراسة تجريبية لحلول الظلم القائمة على الخادم لسيناريو عبر البروتوكولات للبث التكيفي عبر HTTP/3 و HTTP/2

تشانه مينه تران* ثو نجوين دوك فان شوان تان إيجي كاميوكا

كلية الدراسات العليا في الهندسة والعلوم ، معهد شيبورا للتكنولوجيا ، اليابان.

الخلاصة:
منذ إدخال HTTP / 3، ركز البحث على تقييم تأثيره على البث التكيفي الحالي عبر HTTP (HAS). من بين هذه الأبحاث ، نظرًا لبروتوكولات النقل غير ذات الصلة ، حظي الظلم عبر البروتوكولات بين HAS عبر HTTP / 3 (HAS / 3) و HAS عبر HTTP / 2 (HAS / 2) باهتمام كبير. لقد وجد أن عملاء HAS / 3 يميلون إلى طلب معدلات بت أعلى من عملاء HAS / 2 لأن النقل QUIC يحصل على عرض نطاق ترددي أعلى لعملائه HAS / 3 من TCP لعملائه HAS / 2. نظرًا لأن المشكلة تنشأ من طبقة النقل ، فمن المحتمل أن حلول الظلم المستندة إلى الخادم يمكن أن تساعد العملاء في التغلب على مثل هذه المشكلة. لذلك ، في هذه الورقة ، تم إجراء دراسة تجريبية لحلول الظلم القائمة على الخادم لسيناريو البروتوكول المتقاطع لـ HAS / 3 و HAS / 2. تظهر النتائج أنه على الرغم من فشل حل توجيه معدل البت في مساعدة العملاء على تحقيق العدالة ، فإن حل تخصيص النطاق الترددي يوفر أداءً فائقًا.

الكلمات الرئيسية: دفق تكيفي ، عبر بروتوكول ، HTTP/3، QUIC ، خادم ، غير منصفة.