Does TCP New Congestion Window Validation Improve HTTP Adaptive Streaming Performance?

Mihail Yanev
University of Glasgow
Glasgow, UK
m.yanev.1@research.gla.ac.uk

Stephen McQuistin
University of Glasgow
Glasgow, UK
sm@smcquistin.uk

Colin Perkins
University of Glasgow
Glasgow, UK
csp@csperkins.org

ABSTRACT

When using HTTP adaptive streaming, video traffic exhibits on-off behaviour with frequent idle periods that can interact poorly with TCP congestion control algorithms. New congestion window validation (New CWV) modifies TCP to allow senders to recover their sending rate more quickly after certain idle periods. While previous work has shown that New CWV can improve *transport* performance for streaming video, it has not been shown if this translates to improved *application* performance in terms of playback stability. In this paper, we show that New CWV can reduce video re-buffering events by up to 4%, and limit representation switches by 12%, without any changes to existing rate adaptation algorithms.

CCS CONCEPTS

• **Networks** → *Transport protocols*; *Application layer protocols*; Network simulations; **Network control algorithms**.

KEYWORDS

TCP, MPEG DASH, Congestion Window Validation

ACM Reference Format:
Mihail Yanev, Stephen McQuistin, and Colin Perkins. 2022. Does TCP New Congestion Window Validation Improve HTTP Adaptive Streaming Performance?. In *The 32nd edition of the Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'22), June 17, 2022, Athlone, Ireland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3534088.3534347

NOSSDAV'22, June 17, 2022, Athlone, Ireland

ACM ISBN 978-1-4503-9383-6/22/06…$15.00
https://doi.org/10.1145/3534088.3534347

1 INTRODUCTION

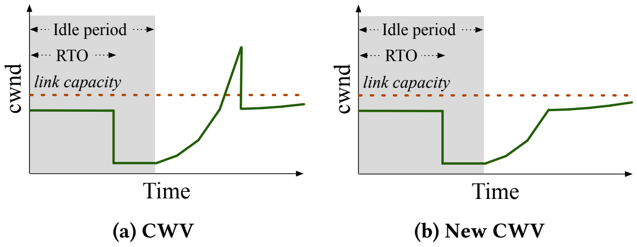
Video streaming over HTTP is commonplace, and comprises the majority of Internet traffic [21]. The performance of such HTTP adaptive streaming video is generally very good, and gives a high-quality user experience.

There are, however, some scenarios where HTTP adaptive streaming can perform comparatively poorly [14, 25]. In particular, interactions between the on-off traffic patterns generated by adaptive streaming applications and TCP congestion control can reduce the performance of throughput-based video rate adaptation schemes [2, 26]. In some cases, this is due to the TCP congestion window validation (CWV) [20] algorithm that, while preventing TCP clients from sending using stale knowledge of the network, has been shown to negatively impact throughput of rate-limited applications [17], including adaptive streaming. New congestion window validation (New CWV) [7] has been proposed to address this. Prior work [17] demonstrated that New CWV has the desired *transport* layer impact, but it remains to show if this translates to improved *application* quality of experience (QoE).

In this paper, we investigate whether enabling New CWV improves video playback stability and, more generally, video QoE for HTTP adaptive streaming over TCP. We compare video performance using TCP New Reno with the CWV and New CWV algorithms. We collect standard video performance metrics, including bit-rate oscillation, and stall time, to measure stability and QoE. Further, to quantify the impact of New CWV with respect to the inferred network state at the client, we also record the immediate and smoothened client's link capacity estimations for each delivered chunk.

We make the following contributions: (i) we implement New CWV for Linux kernel 5.4; (ii) we evaluate the application layer impact of New CWV; and (iii) we demonstrate that New CWV improves video stability, with a 12% reduction in bit rate switches, and a 4% reduction in rebuffering time.

To the best of our knowledge, we are the first to study the application layer impact of enabling New CWV. Nazir et al. [17] demonstrated the effect of New CWV on the transport layer, and there has been much work on application layer rate adaptation [9, 16, 29]. In contrast, we only change the transport, leaving the application as is, to study the transport impact on the application. Improving performance via



(a) CWV

(b) New CWV

Figure 1: *cwnd* growth following an idle period

transport layer modifications could allow for simpler rate adaptation algorithms at the application layer.

We structure the remainder of this paper as follows. In Section 2, we introduce TCP congestion window validation, including its limitations with respect to HTTP adaptive video flows, before describing New CWV. Section 3 describes our experimental setup, and the transport and application layer impact of enabling New CWV. Section 4 describes related work, and Section 5 concludes.

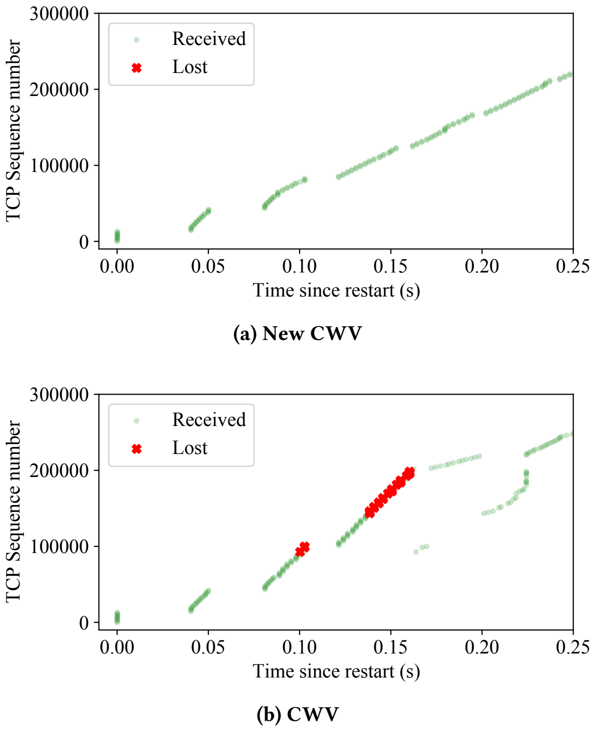
2 CONGESTION WINDOW VALIDATION

In HTTP adaptive streaming, a server provides pre-encoded video chunks in different representations, each encoded at multiple bit rates. The client use a rate adaptation algorithm to determine the best representation to request at any given time. The goal of the client is to maximise QoE within the network's capacity. This can be a challenge since different, often contradictory, QoE heuristics need be considered [23].

Throughput-based rate adaptation algorithms require a stable and accurate estimate of TCP throughput. However, interactions between the on-off traffic pattern of streaming video and TCP congestion control can cause significant fluctuations in TCP behaviour, impacting performance.

In particular, during idle periods in between transmission of video chunks, the TCP congestion controller's knowledge of the network capacity becomes stale. To avoid sending with a possibly unrepresentative congestion window, once the link has been idle for a period longer than the retransmission timeout, T_{rto}, the congestion window validation [20] (CWV) algorithm resets the TCP congestion window (*cwnd*) to its initial value and forces the connection to re-enter slow-start. Figure 1a illustrates this behaviour. CWV has become standard practice [4], and is enabled by default in the latest stable Linux kernel (5.4). This slow start after idle, however, causes bursts of packet loss when the *cwnd* overshoots the link capacity at the end of the slow start (Figure 2b). This can to interact poorly with HTTP adaptive streaming and other application-limited transmissions [6].

To address this, new congestion window validation [7] was proposed. During slow start after idle, rather than relying on packet loss to re-discover its appropriate *cwnd* value, New CWV preserves the *cwnd* before the idle period and later uses it as the slow start threshold (*ssthresh*). That is, it exits the slow-start phase before loss occurs, assuming no change in available bandwidth. Figure 1b shows the growth of *cwnd* following an idle period under New CWV.



(a) New CWV

(b) CWV

Figure 2: Resumption after an idle period

We implemented New CWV in Linux[1]. We used [22] as a base, which provides an implementation for Linux 3.18, and ported the changes to Linux 5.4. Our implementation altered two files, adding 143 and removing 49 lines of code.

We use this to better illustrate the impact of New CWV on flows restarting after an idle period. The New CWV connection (Figure 2a) uses the previously set *ssthresh* value and leaves slow-start early, avoiding packet loss, after reaching the *ssthresh* in the third flight of packets after restarting. If the same connection used CWV, the senders would not preserve the *ssthresh* and would rely on loss to exit slow-start, as seen at the end of the third and fourth flights of packets in Figure 2b. In this example, the connection using CWV enters congestion avoidance ~160ms after transmission restarts, while that using New CWV can do so ~50ms earlier and avoid

[1] A snapshot of our implementation, as used in this paper, is available at http://dx.doi.org/10.5525/gla.researchdata.1278, while the main repository for the work is at https://github.com/glasgow-ipl/newcwv-nossdav2022.

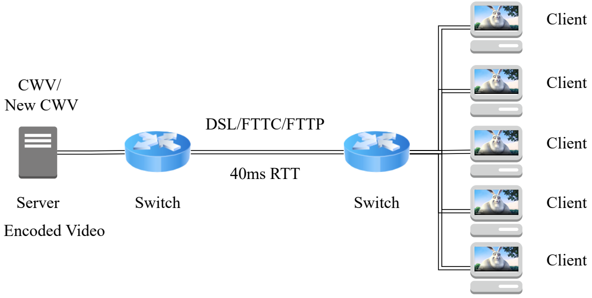


Figure 3: Experimental Setup

packet loss. Overall, New CWV results in fewer lost packets, and returns to its previous sending rate without overshoot after loss, giving more predictable transmission.

New CWV has been shown to improve *transport* layer performance of rate-limited applications when compared with CWV [17], and the results we have presented here validate that, although it remains to show how this translates into *application* layer performance [25].

3 EVALUATING NEW CWV FOR VIDEO

In this section, we first validate the results of Nazir et al. [17], before testing whether New CWV will enable *applications* to obtain more consistent throughput estimates, and, in turn, improve the stability of rate adaptation algorithms. We first describe our experimental setup (§3.1). Second, we investigate the transport layer impact to check if New CWV connections obtain more consistent throughput estimates (§3.2). Then, we investigate the application impact, specifically the difference on video QoE due to New CWV (§3.3). Finally, we summarise our findings (§3.4).

3.1 Experimental Setup

Our evaluation consists of a network emulated in Mininet, running on Ubuntu 20.04, as shown in Figure 3. Both the server and clients use TCP New Reno and run Linux Kernel 5.4.0 modified to include New CWV and RFC 3339 [18] compliant timestamps to enable high precision event tracking.

The server uses `nginx` (v1.18) with HTTP/2, serving three representations of Big Buck Bunny[2]: 480p (0.44Mbps), 720p (2.64Mbps), and 1080p (4.82Mbps). Each is encoded in chunks of 3 seconds duration. Clients use Firefox (v91) with `dash.js`[3] (v4.0.0). We report results with both the THROUGHPUT and DYNAMIC [24] algorithms implemented in `dash.js`.

The network is configured with 40ms RTT, matching a typical RTT to a nearby CDN node. Router queues are sized to the bandwidth delay product. Three different bandwidth profiles are evaluated, representing ADSL2 (10Mbps), FTTC (50Mbps), and FTTP (145Mbps) links. We show the results for ADSL2 and FTTC links. While FTTP links are common in some regions, ADSL2 and FTTC links are still widely used [1, 5, 8, 19]. As higher resolution video and other network-heavy operations such as virtual reality environments see more use we expect issues observed with ADSL2 and FTTC will translate to the higher capacity FTTP links. We send only video traffic and assume consistent RTT, broadly modelling a home environment with multiple people watching simultaneous video streams and little other traffic. Evaluating effects of cross traffic and video synchronisation is future work.

To evaluate the impact of congestion and competing flows, experiments were run with both DYNAMIC and THROUGHPUT algorithms and several (1, 2, 3, and 5) simultaneous clients. Finally, to reduce noise, we ran each combination of CWV or New CWV, adaptive algorithm, number of clients, and link type, 10 times before reporting the average results. The results presented includes data accumulated from 480 simulations (2 congestion control algorithms × 2 adaptive algorithms × 3 link types × 4 client variations × 10 repetitions). During each run we collect client bandwidth estimates. Additionally, to evaluate the video QoE impact we report the rebuffer ratio and bit-rate switch frequency distribution.

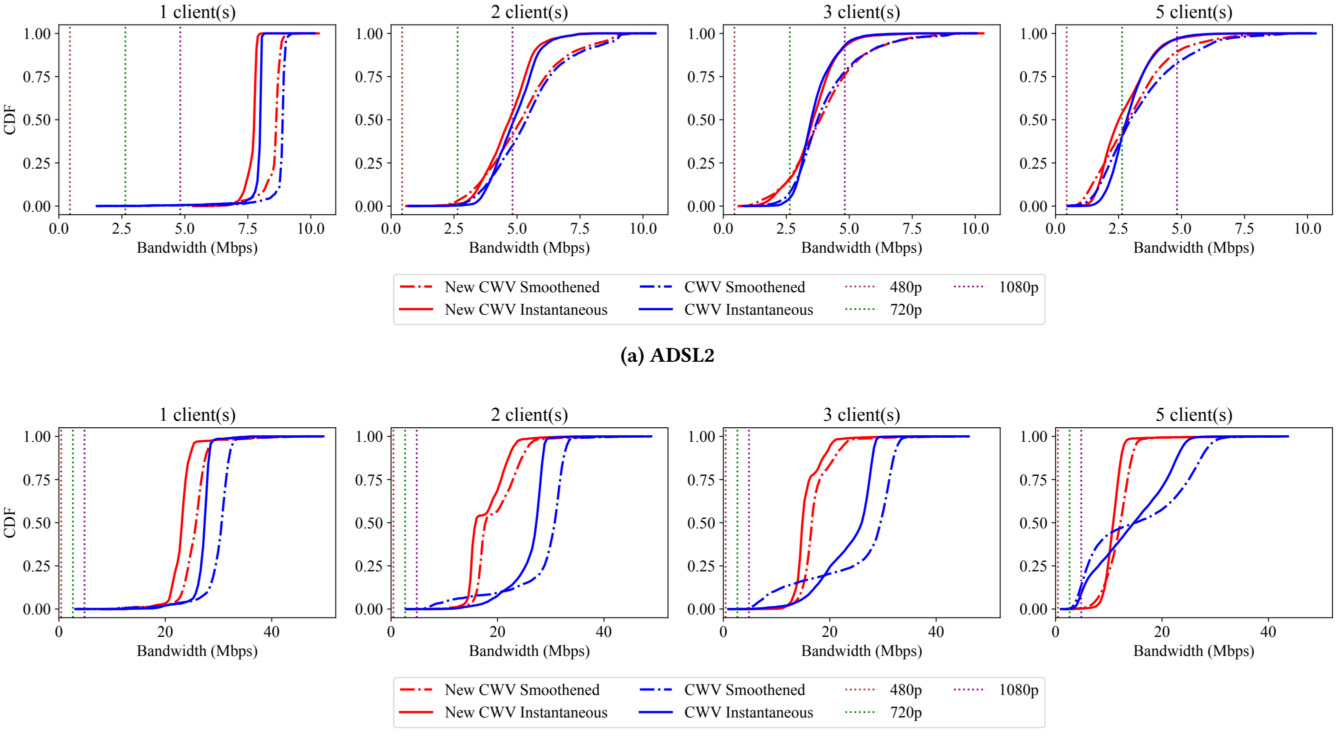
3.2 Impact on Transport Performance

New CWV alters TCP *cwnd* sizing behaviour, allowing it to recover more quickly after idle periods, and, as shown in Section 2, prevents packet loss when exiting slow start which should give more stable bandwidth estimates.

To evaluate this, we collect instantaneous and smoothed bandwidth estimates from the clients. The instantaneous estimate is obtained by dividing the size of the chunk by the time taken to download it. The smoothed estimate is a function of the instantaneous estimate and other variables, f.e., historical measurement data and safety or "dampening" factors. The former is the throughput measurement as seen by the end-point, while the latter is the input value to the client's rate adaptation algorithm.

The cumulative distribution of instantaneous and smoothed throughput is shown in Figure 4. The ADSL2 results, Fig. 4a, are similar for CWV and New CWV. This is due to the relatively low link capacity and the exponential increase of the slow-start not being able to overshoot too far over the capacity after an idle period, so avoiding loss of many packets and matching New CWV behaviour. The FTTC results, Fig. 4b, show a consistently tighter distribution with New CWV than CWV, representing a more stable throughput estimate. The exception is when only one client is used, where the link capacity was much higher than the total video bandwidth that could be requested; preventing the client from

[2] https://download.blender.org/demo/movies/BBB/
[3] https://github.com/Dash-Industry-Forum/dash.js



(a) ADSL2

(b) FTTC

Figure 4: Measured dash.js client bandwidth estimates DYNAMIC

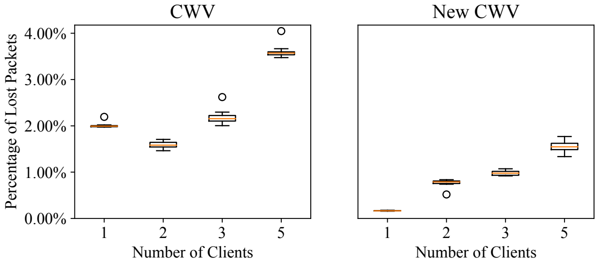


Figure 5: Lost Packets DSL DYNAMIC

obtaining an accurate estimation of the available capacity. FTTP results are not shown, since in no case was the link capacity reached, and there were no significant difference in performance between CWV and New CWV.

In addition to being more consistent, Clients with New CWV enabled reported throughput estimates that were lower overall. This is matches the results in Section 2, showing flows using New CWV leaving slow start earlier. Nazir et al. [17] also saw this behaviour. This also supports our initial hypothesis that streaming clients measuring throughput will be able to obtain estimates that are more stable.

We illustrate the impact of New CWV on packet loss in Figure 5. New CWV consistently achieves lower loss compared to CWV, with the former having packet loss rates that are less than half the latter. This confirms our hypothesis for loss, and matches Section 2: New CWV exits slow-start earlier, does not overshoot its window, and therefore is able to avoid loss at the end of slow-start.

The results in this section are reported for the DYNAMIC rate adaptation algorithm. Results for the THROUGHPUT algorithm show no significant differences.

3.3 Impact on Video QoE

To evaluate whether improved New CWV transport layer performance translates into improved application QoE, we explore impact on application-level metrics such as rebuffer ratio and bit rate switch frequency.

In all cases, we report results for ADSL2 only. This link type, in combination with the video encodings used, best highlights scenarios where New CWV is beneficial. With FTTP links, all clients were able to stream at the highest quality without hitting the link limits. This was also the case for the FTTC links, with the exception of the 5 client scenario where we observed a similar pattern as that shown for

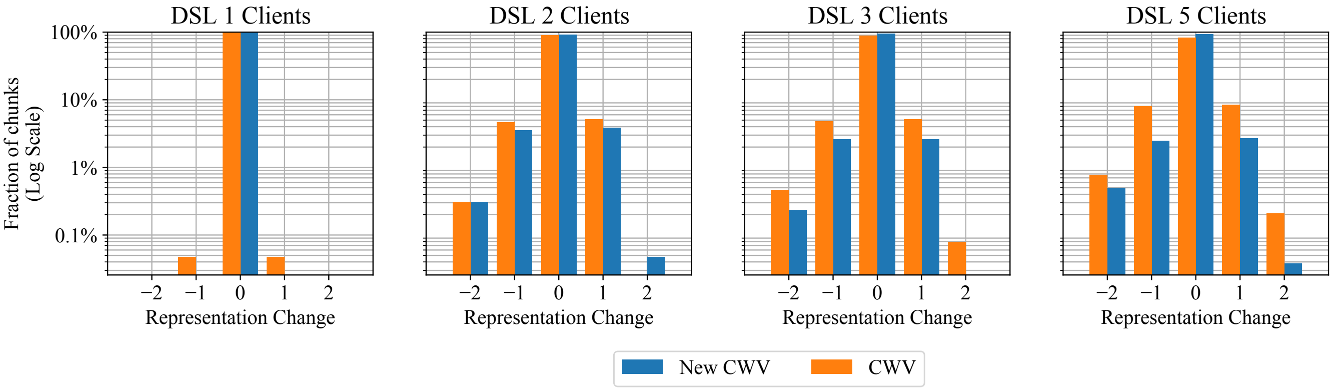


Figure 6: Absolute Bitrate Switches (THROUGHPUT)

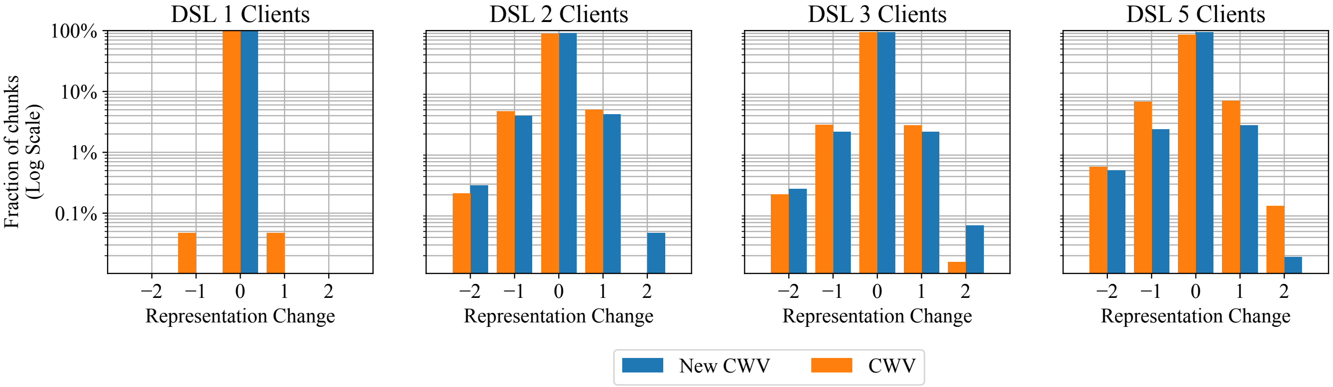


Figure 7: Absolute Bitrate Switches (DYNAMIC)

ADSL2. Congestion control changes, such those we discuss, are ineffective when video flows are application limited and cannot cause congestion. Future higher-bandwidth applications, such as higher quality video, AR, or VR, will shift the congestion threshold upwards.

Figure 6 shows the bit rate switch distribution for the THROUGHPUT algorithm; Figure 7 shows DYNAMIC.[4] In the extreme, with 5 clients, connections with CWV experience representation changes between 17.4% of video chunks, compared to 5.7% for connections with New CWV. Fewer clients congest the link less, and see correspondingly fewer shifts.

Comparing Figures 6 and 7, we see that when CWV is used, DYNAMIC shows a significant reduction in representation changes compared to THROUGHPUT. Connections using New CWV exhibit fewer representations shifts, irrespective of the rate adaptation algorithm.

Figure 8 shows the fraction of the video playback where the connection has stalled when using the THROUGHPUT algorithm. The value is shown in percentage relative to the whole media playtime. We see that New CWV experiences less rebuffering overall. The mean rebuffering values for the 5 client case are 5% and 1.5% for CWV and New CWV respectively, and correspondingly less for fewer clients. Figure 9 shows results for DYNAMIC, and again we see that the performance of the connections using CWV has improved, rebuffer-ratio is lower. Performance of New CWV connections does not differ significantly from what was already observed with THROUGHPUT, and is still better than the performance with CWV even in cases where DYNAMIC was used.

We do not present average bit rate results, due to space limitations, but these show little variation between CWV and New CWV. Differences that exist show slightly improved stability for New CWV, consistent with our other results.

[4] The representation change (*x-axis*) is the magnitude of the chunk-by-chunk bit rate switches. For example, if a chunk is requested at the same bit rate as the preceding chunk, the difference is zero. If the chunk is of the next higher encoding, compared to its predecessor, the difference is 1, and if it is of the next lower encoding the difference is -1, and so on.

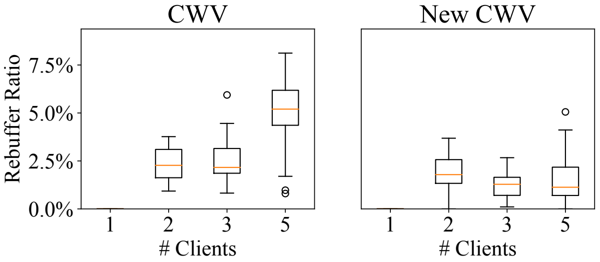


Figure 8: DSL Rebuffer Ratio (THROUGHPUT)

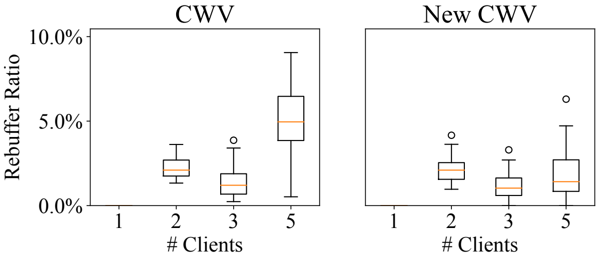


Figure 9: DSL Rebuffer Ratio (DYNAMIC)

Overall, New CWV achieves higher video stability when multiple clients are competing on a constrained link, with improved encoding stability and decreased rebuffering time. We observe that when New CWV is enabled, application metrics achieved for connections using either DYNAMIC and THROUGHPUT are very similar. A more complex rate adaptation algorithm, such as DYNAMIC, has a significant impact on connections using CWV, but still gives lower QoE than the simpler adaptation algorithm used with New CWV.

3.4 Summary

The use of New CWV contributes to more consistent bandwidth estimates (Figure 4) and leads to fewer lost packets (Figure 5), fewer representation switches (Figures 6 & 7), and lower rebuffering ratio (Figures 8 & 9). This is since the more consistent measurements better match the available network conditions, allowing clients using New CWV to better adapt to the constraints of the bottleneck link.

The scenarios investigated by this work have used links that emulated a wired residential Internet connection. WiFi and cellular links, however, are known to have very different properties that can affect TCP performance. Understanding how use of TCP New CWV affects video performance on such links is outside the scope of this current work.

Our experiments investigated the effects of New CWV on video delivered using TCP New Reno. This was done to align with [17], and because a recent study [15] found New Reno was still widely used for video delivery. TCP Cubic grows its congestion window in slow start using the HyStart++ algorithm [3], and TCP BBRv2 tracks bandwidth estimates during slow start, in both cases to avoid the overshoot problems we discuss in Section 2 and improve stability. These can potentially benefit from retaining state after idle periods in the manner of New CWV; exploring this is for future work.

We consider similar scenarios to [17], but our work was built to match scenarios that could be seen in 2022. Our findings broadly match theirs, consistent with the changes in HTTP adaptive streaming since their study.

4 RELATED WORK

New CWV enhances the CWV algorithm [20]. Both address problems with resumption after an idle period when TCP has a stale view of the network. CWV addresses the issue for bulk, network-limited applications; New CWV improves the algorithm for rate-limited applications. As noted above, extensions such as HyStart++ [3], also achieve similar results..

A wide variety of ABR algorithms have been proposed, providing longer-term rate adaptation to complement TCP dynamics (e.g., [9, 12, 24, 25, 27, 28]). Work on such ABR algorithms has slowed, however, with more recent proposals optimising for specific use cases (e.g., [13]), perhaps due to the diminishing improvements obtained [29]. We identified cases (e.g., multiple clients competing on a constrained link) where the current state-of-the art algorithms perform poorly, and showed that transport changes, such as enabling New CWV, can have positive QoE impact, with up to 4% points of improved rebuffering and 12% points of more stable chunk selection. It is difficult to deploy changes to TCP, but the introduction of QUIC [11] provides an opportunity to change the transport. QUIC congestion control states that New CWV *may* be implemented [10]; our results suggest it *should* be implemented when using QUIC with ABR video.

5 CONCLUSIONS

New CWV improves HTTP adaptive streaming playback stability. We compared video delivery with CWV and New CWV, and validated the results shown by previous work [17]. Enabling New CWV, a transport layer change, can improve application layer performance, reducing the number of encoding switches by up to 12% points and rebuffering time by up to 4% points, improving QoE and making transmission more predictable. These improvements help simple rate adaptation algorithms compete with the state of the art.

ACKNOWLEDGEMENTS

This work is supported by the UK Engineering and Physical Sciences Research Council.

REFERENCES

[1] ACCC. 2022. Measuring Broadband Australia program. https://www.accc.gov.au/regulated-infrastructure/communications/monitoring-reporting/measuring-broadband-australia-program. (2022).

[2] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen. 2012. What happens when HTTP adaptive streaming players compete for bandwidth?. In *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, Toronto, Canada, 9–14. https://doi.org/10.1145/2229087.2229092

[3] P. Balasubramanian, Y. Huang, and M. Olson. 2022. HyStart++: Modified Slow Start for TCP. Internet Engineering Task Force. (Jan. 2022). Work in progress.

[4] E. Blanton, V. Paxson, and M. Allman. 2009. TCP Congestion Control. Internet Engineering Task Force. (Sept. 2009). RFC 5681.

[5] EC. 2022. Broadband Connectivity in the Digital Economy and Society Index. https://digital-strategy.ec.europa.eu/en/policies/desi-connectivity. (2022).

[6] J. Esteban, S. A. Benno, A. Beck, Y. Guo, V. Hilt, and I. Rimac. 2012. Interactions between HTTP adaptive streaming and TCP. In *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, Toronto, Canada, 21–26. https://doi.org/10.1145/2229087.2229094

[7] G. Fairhurst, S. Sathiaseelan, and R. Secchi. 2015. Updating TCP to Support Rate-Limited Traffic. Internet Engineering Task Force. (Oct. 2015). RFC 7661.

[8] FCC. 2022. Eleventh Measuring Broadband America Fixed Broadband Report A Report on Consumer Fixed Broadband Performance in the United States Office of Engineering and Technology. (2022).

[9] T. Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson. 2015. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. *Computer Communication Review* 44, 4 (Feb. 2015), 187–198. https://doi.org/10.1145/2619239.2626296

[10] J. Iyengar and I. Swett. 2021. QUIC Loss Detection and Congestion Control. Internet Engineering Task Force. (May 2021). RFC 9002.

[11] J. Iyengar and M. Thomson. 2021. QUIC: A UDP-Based Multiplexed and Secure Transport. Internet Engineering Task Force. (May 2021). RFC 9000.

[12] J. Jiang, V. Sekar, and H. Zhang. 2014. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking* 22, 1 (Feb. 2014), 326–340. https://doi.org/10.1109/TNET.2013.2291681

[13] T. Karagkioules, R. Mekuria, D. Griffioen, and A. Wagenaar. 2020. Online learning for low-latency adaptive streaming. In *Proceedings of the Multimedia Systems Conference*. ACM, Online, 315–320. https://doi.org/10.1145/3339825.3397042

[14] J. Kua, G. Armitage, and P. Branch. 2017. A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP. *IEEE Communications Surveys and Tutorials* 19, 3 (July 2017), 1842–1866. https://doi.org/10.1109/COMST.2017.2685630

[15] A. Mishra, X. Sun, At. Jain, S. Pande, R. Joshi, and B. Leong. 2019. The Great Internet TCP Congestion Control Census. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3 (Dec. 2019), 1–24. Issue 3. https://doi.org/10.1145/3366693

[16] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang. 2012. QDASH: A QoE-aware DASH system. In *Proceedings of the Multimedia Systems Conference*. ACM, Chapel Hill, NC, USA, 11–22. https://doi.org/10.1145/2155555.2155558

[17] S. Nazir, Z. Hossain, R. Secchi, M. Broadbent, A. Petlund, and G. Fairhurst. 2014. Performance Evaluation of Congestion Window Validation for DASH Transport. In *Proceedings of the Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, Singapore, 67–72. https://doi.org/10.1145/2597176.2578275

[18] C. Newman and G Klyne. 2002. Date and Time on the Internet: Timestamps. Internet Engineering Task Force. (July 2002). RFC 3339.

[19] Ofcom. 2020. UK Home Broadband Performance: Technical Report. (May 2020). https://www.ofcom.org.uk/__data/assets/pdf_file/0038/194897/uk-home-broadband-performance.pdf

[20] H. Padhye, S. Floyd, and M. Handley. 2000. TCP Congestion Window Validation. Internet Engineering Task Force. (June 2000). RFC 2861.

[21] Sandvine. 2019. The Global Internet Phenomena Report. https://www.sandvine.com/global-internet-phenomena-report-2019. (Sept. 2019).

[22] R Secchi. 2016. Linux New CWV kernel module. https://github.com/rsecchi/newcwv. (March 2016).

[23] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia. 2015. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys and Tutorials* 17, 1 (Jan. 2015), 469–492. https://doi.org/10.1109/COMST.2014.2360940

[24] K. Spiteri, R. Sitaraman, and D. Sparacio. 2019. From theory to practice: Improving Bitrate adaptation in the DASH reference player. *ACM Transactions on Multimedia Computing, Communications, and Applications* 15 (Aug. 2019), 1–29. Issue 25. https://doi.org/10.1145/3336497

[25] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. 2016. BOLA: Near-optimal bitrate adaptation for online videos. In *Proceedings of the Infocom Conference*. IEEE, San Francisco, CA, USA, 1–9. https://doi.org/10.1109/INFOCOM.2016.7524428

[26] D. Stohr, A. Frömmgen, A. Rizk, M. Zink, R. Steinmetz, and W. Effelsberg. 2017. Where are the sweet spots? A systematic approach to reproducible DASH player comparisons. In *Proceedings of the International Conference on Multimedia*. ACM, Mountain View, CA, USA, 1113–1121. https://doi.org/10.1145/3123266.3123426

[27] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli. 2016. CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the SIGCOMM Conference*. ACM, Florianopolis, Brazil, 272–285. https://doi.org/10.1145/2934872.2934898

[28] C. Wang, A. Rizk, and M. Zink. 2016. SQUAD: A Spectrum-based Quality Adaptation for Dynamic Adaptive Streaming over HTTP. In *Proceedings of the Multimedia Systems Conference*. ACM, Klagenfurt, Austria, 1–12. https://doi.org/10.1145/2910017.2910593

[29] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. *Computer Communication Review* 45, 4 (Aug. 2015), 325–338. https://doi.org/10.1145/2785956.2787486