Manus Manum Lavat: Media Clients and Servers Cooperating with Common Media Client/Server Data

Ali C. Begen
Ozyegin University
Istanbul, Turkey

ABSTRACT

The newly rectified CTA standard — Common Media Client Data (CMCD) — allows content providers to get insights into the performance of their large-scale streaming operations. Its sister standard — Common Media Server Data (CMSD) — is in the works and will allow servers to send hints to other servers and clients. The CMCD/CMSD combo is the long-awaited upgrade to HTTP adaptive streaming systems.

CCS CONCEPTS

• **Networks** → **Application layer protocols**; • **Information systems** → *Multimedia streaming*.

KEYWORDS

CMCD, CMSD, adaptive streaming, CDN, OTT, DASH, HLS.

ACM Reference Format:
Ali C. Begen. 2021. Manus Manum Lavat: Media Clients and Servers Cooperating with Common Media Client/Server Data. In *Applied Networking Research Workshop (ANRW '21), July 24–30, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3472305.3472886

1 OVERVIEW

The most common technology for media delivery is HTTP adaptive streaming (HAS) [8, 9, 15] using segmented media. The key objective in HAS is to deliver media at high quality with zero interruptions [3, 20]. Content delivery networks (CDNs), which are special caching and distribution overlays, play an important role in supporting the massive demand in HAS services all around the world. CDNs use a large number of inter-connected caches at the edge and provide low-latency access to live and on-demand content segments.

ANRW '21, July 24–30, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8618-0/21/07…$15.00
https://doi.org/10.1145/3472305.3472886

HAS is a complex technology with dynamics that need to be studied thoroughly. The experience from the deployments in the last 10+ years suggests that streaming clients typically operate in an unfettered greedy mode and they are not necessarily designed to behave well in environments where other clients exist or network conditions can change dramatically. This largely stems from the fact that clients make only indirect observations at the application (HTTP) layer (and limitedly at the transport layer, if any at all) [4].

Typically, there are three primary camps when it comes to scaling and improving streaming systems: (*i*) servers control client's behavior/actions and the network uses appropriate QoS, (*ii*) servers and clients cooperate with each other and/or the network, or (*iii*) clients stay in control and no cooperation with the servers or network is needed as long as there is *enough* capacity in the network (said differently, use dumb servers and network and throw more bandwidth at the problem). Each of these have its own merits and none of them is a one-size-fits-all solution. To date, the more common view in the streaming world has been more or less aligned with the third camp above. In line with this, limited supporting evidence [27] showed a strictly controlled system could perform worse if HAS traffic were treated differently from the non-HAS traffic. Nonetheless, some of the networking equipment vendors and service providers have been historically closer to the first camp above since they viewed the concept of streaming video in an uncontrolled – or loosely controlled – fashion as flawed. These vendors and providers were accustomed to old methods that heavily benefited from using QoS and simply thought non-network-empowered HAS could never beat the managed video delivery. Later on, it became nearly certain that HAS would be the preferred delivery approach for both over-the-top (OTT) and managed content – not just for the former. The market transition that took place in the voice world was doomed to repeat itself in the video world.

Almost all streaming methods treat the network as a "black box" as the network (or a server) does not usually offer anything beyond carrying the bits (or responding to client's requests). Through running an internal rate-adaptation logic, each streaming client selfishly tries to grab a portion of the network/server capacity. While doing so, clients compete for bandwidth with each other in addition to other co-existing non-HAS flows. Clients make indirect observations of the network state, subsequently rate-adapt and this cycle continues till the end of the streaming session. In practice, one can never have a capacity that is *enough* for everything on the server, ISP aggregation link, access link or in-home Wi-Fi network. That is, there is always someone else streaming from the same server or a different server but still sharing some part of the network path, downloading a file, surfing the web or otherwise simultaneously consuming some of the available bandwidth. Thus, the premise for the third camp above rarely holds. Due to the contention, problems such as bitrate oscillations, network resource underutilization, sustained starvation and (bitrate) unfairness among the clients are inevitable [4, 10]. Following suit of studies such as [5] and [21], researchers and developers started showing more interest in finding a compromise between the first and third camp and looking into bidirectional hinting between the clients, servers and network.

Intuitively, using hints should improve streaming since it helps the clients and servers take more appropriate actions. The improvement could be in terms of better viewer experience and supporting more viewers for the given amount of network resources, or the added capability to explicitly support *controlled unfairness* (as opposed to bitrate fairness) based on features such as content type, viewer profile and display characteristics [7]. The first concrete proposal for defining a control plane framework and enabling a message exchanging platform between different streaming elements [2] was presented in a workshop jointly organized by the MPEG and IETF in 2013 [25]. This proposal established the foundation for the Server and Network Assisted DASH (SAND) standard [1] that was published as ISO/IEC 23009-5 in 2017 after two years of development. Generally speaking, information exchange is useful. It is most useful, however, when the information is relevant, actionable and up-to-date. Thus, in a system using the SAND concepts, an important question is what information is relevant and actionable. After the standard's publication, many studies (*e.g.*, [11–13, 16, 22, 24, 26, 28–30]) focused on this question and offered a number of answers.

For several years, the CDN providers hardly embraced the SAND concept and showed almost no interest until 2019 when the Consumer Technology Association (CTA) started a new initiative to address a long-standing problem for them: how could a streaming client relay media (*e.g.*, segment type/duration/format, content/session IDs) and playback (*e.g.*, current buffer length and latency) related information so that the CDN could tie the individual GET requests to playback sessions, harmonize its and client's logs to accurately generate dashboard metrics such as delivery performance, player software issues and viewer experience, and react to the time constraints implicit in media segment requests (*e.g.*, prioritize delivery for *urgent* requests). CTA published this Common Media Client Data (CMCD; CTA-5004) [17] spec in Sept. 2020. The first evaluation results are already out [14] and many other studies are underway.

During the development of CMCD, an issue [18] was raised about the possibility of sending meta information and hints from the CDN to the streaming clients. The discussions eventually led to the concept of Common Media Server Data (CMSD), which will be developed as a companion standard to CTA-5004. The working group is currently in the phase of collecting use cases. The hints considered in these initial use cases include server-side bandwidth estimates, limits for the playback bitrate, redirection suggestions, caching indications, breadcrumb data and server/network load signals. Figure 1 illustrates a media distribution system that uses CMCD and CMSD together.

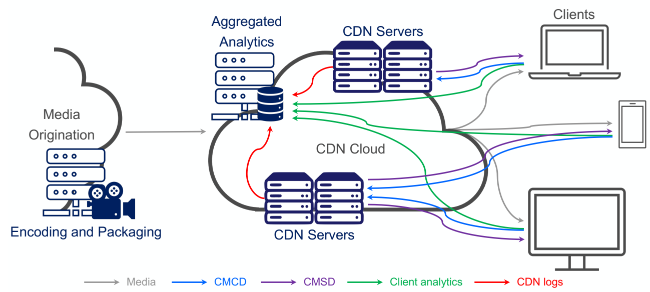

Figure 1: Client-CDN cooperation with CMCD/CMSD.

CMSD can help in many ways. A client that would mistakenly downshift due to misinterpretation of a cache miss can be warned via CMSD. Another client that would normally start the session with the lowest-bitrate segments can be hinted to fetch higher-bitrate segments. Oscillating clients [6] can be assisted and easily stabilized via CMSD. Since storage capacity is limited in practice, not all encodings can be cached and rate-adapting clients can get confused in certain circumstances [23]. CMSD also helps in this case using caching indications to list what is cached or not cached letting the clients make more informed decisions.

2 CALL FOR ACTION

The goal of this lightning paper is to lay out observations about where things stand so that we can rationally evaluate what to include in the new CMSD spec (and future versions of CMCD). It goes without saying that the more streaming clients and CDNs adopt CMCD/CMSD, the greater the mutual benefits are. All interested parties should get involved by contributing to the spec, working on an implementation or running a trial. One can join the Web Application Video Ecosystem (WAVE) project in CTA or simply follow the public GitHub issue tracker [19].

REFERENCES

[1] ISO/IEC 23009-5:2017 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 5: Server and network assisted DASH (SAND). [Online] Available: https://www.iso.org/standard/69079.html. Accessed on Feb. 20, 2021.

[2] m30355: Contribution to SAND CE. [Online] Available: https://dms.mpeg.expert/doc_end_user/documents/105_Vienna/wg11/m30355-v2-m30355-SAND-r1.zip. Accessed on May 5, 2021.

[3] A. A. Barakabitze et al. QoE management of multimedia streaming services in future networks: a tutorial and survey. *IEEE Communications Surveys & Tutorials*, 22(1):526–565, 2020.

[4] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis. What happens when HTTP adaptive streaming players compete for bandwidth? In *ACM NOSSDAV*, 2012 (DOI: 10.1145/2229087.2229092).

[5] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen. Server-based traffic shaping for stabilizing oscillating adaptive streaming players. In *ACM NOSSDAV*, 2013 (DOI: 10.1145/2460782.2460786).

[6] S. Akhshabi, A. C. Begen, and C. Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP. In *ACM MMSys*, 2011 (DOI: 10.1145/1943552.1943574).

[7] A. C. Begen. Spending quality time with the web video. *IEEE Internet Comput.*, 20(6):42–48, Nov./Dec. 2016 (DOI: 10.1109/MIC.2016.49).

[8] A. C. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 1: Streaming protocols. *IEEE Internet Computing*, 15(2):54–63, 2011 (DOI: 10.1109/MIC.2010.155).

[9] A. C. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 2: Applications, standardization, and open issues. *IEEE Internet Computing*, 15(3):59–63, 2011 (DOI: 10.1109/MIC.2010.156).

[10] A. C. Begen, T. Donahue, J. Gahm, and Z. Li. Challenges with large-scale adaptive streaming deployments. In *SCTE Cable-Tec Expo*, 2014.

[11] A. Bentaleb, A. C. Begen, and R. Zimmermann. SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking. In *ACM Multimedia*, 2016 (DOI: 10.1145/2964284.2964332).

[12] A. Bentaleb, A. C. Begen, and R. Zimmermann. QoE-aware bandwidth broker for HTTP adaptive streaming flows in an SDN-enabled HFC network. *IEEE Trans. Broadcasting*, 64(2):575–589, 2018 (DOI: 10.1109/TBC.2018.2816789).

[13] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous. SDNHAS: an SDN-enabled architecture to optimize QoE in HTTP adaptive streaming. *IEEE Trans. Multimedia*, 19(10):2136–2151, 2017 (DOI: 10.1109/TMM.2017.2733344).

[14] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, and R. Zimmermann. Common media client data (CMCD): Initial findings. In *ACM NOSSDAV*, 2021 (DOI: 10.1145/3458306.3461444).

[15] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann. A survey on bitrate adaptation schemes for streaming media over HTTP. *IEEE Communications Surveys & Tutorials*, 21(1):562–585, 2019 (DOI: 10.1109/COMST.2018.2862938).

[16] D. Bhat, A. Rizk, M. Zink, and R. Steinmetz. Network assisted content distribution for adaptive bitrate video streaming. In *ACM MMSys*, 2017.

[17] Consumer Technology Association. CTA-5004: Web Application Video Ecosystem–Common Media Client Data, Sept. 2020.

[18] cta-wave/common-media-client data. No common-media-server-data? [Online] Available: https://github.com/cta-wave/common-media-client-data/issues/19. Accessed on May 5, 2021.

[19] cta-wave/common-media-server data. [Online] Available: https://github.com/cta-wave/common-media-server-data. Accessed on May 5, 2021.

[20] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. In *ACM SIGCOMM*, 2011.

[21] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards network-wide QoE fairness using openflow-assisted adaptive video streaming. In *ACM SIGCOMM Wksp. FhMN*, 2013.

[22] J. W. Kleinrouweler, B. Meixner, and P. Cesar. Improving Video Quality in Crowded Networks Using a DANE. In *ACM NOSSDAV*, 2017.

[23] D. H. Lee, C. Dovrolis, and A. C. Begen. Caching in HTTP adaptive streaming: friend or foe? In *ACM NOSSDAV*, 2014 (DOI: 10.1145/2597176.2578270).

[24] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski. Joint Optimization of QoE and Fairness Through Network Assisted Adaptive Mobile Video Streaming. In *IEEE WiMob*, 2017.

[25] MPEG/IETF. Workshop on session management and control for MPEG DASH. https://mpeg.chiariglione.org/about/events/workshop-session-management-and-control-mpeg-dash. July 2013.

[26] S. Pham, P. Heeren, C. Schmidt, D. Silhavy, and S. Arbanowski. Evaluation of shared resource allocation using SAND for ABR streaming. *ACM TOMM*, 16(2s):1−−18, 2020.

[27] J. Schlack. Managing resources on service provider networks for ABR content. In *3rd Cisco Workshop on Adaptive Media Transport*, 2012.

[28] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, M.-L. Champel, and O. Oyman. Applications and Deployments of Server and Network Assisted DASH (SAND). In *Int. Broadcasting Convention Conf. (IBC)*, 2016 (DOI: 10.1049/ibc.2016.0022).

[29] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey. Enhancing MPEG DASH Performance via Server and Network Assistance. In *Int. Broadcasting Convention Conf. (IBC)*, 2015 (DOI: 10.1049/ibc.2015.0014).

[30] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey. Enhancing MPEG DASH Performance via Server and Network Assistance. *SMPTE Motion Imaging Journal*, 126(1):22–27, 2017 (DOI: 10.5594/JMI.2016.2632338).