Common Media Server Data (CMSD) – Update on Implementations and Validation of Key Use Cases

Stefan Pham[★], Will Law[+], Ali C. Begen[§], Daniel Silhavy[★], Bertrand Berthelot[†], Stefan Arbanowski[★] and Stephan Steglich[★]
[★]Fraunhofer FOKUS, [+]Akamai, [§]Ozyegin University, [†]Broadpeak
[★]Germany, [+]USA, [§]Turkiye, [†]France

ABSTRACT

The CTA-5006 (Common Media Server Data, CMSD) specification establishes a uniform method for media servers to exchange data with each media object response. The aim is to enhance distribution efficiency, performance, and ultimately, the user experience. We provide an overview of CMSD implementations and focus on integrating CMSD into the dash.js reference player. Three use cases are evaluated to demonstrate the advantages of CMSD, including leveraging edge server throughput estimates to improve initial bitrate selection and low-latency live streaming, prefetching manifests and segments to improve startup delay, and allowing an edge server to suggest a playback bitrate to improve the collective experience. The outcomes from the initial implementations confirm the benefits of using CMSD.

CCS CONCEPTS

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

KEYWORDS

CMSD, CMCD, adaptive streaming, CDN, OTT, DASH, HLS, server assistance, network assistance, SAND.

ACM Reference Format:
Stefan Pham[★], Will Law[+], Ali C. Begen[§], Daniel Silhavy[★], Bertrand Berthelot[†], Stefan Arbanowski[★] and Stephan Steglich[★]. 2023. Common Media Server Data (CMSD) – Update on Implementations and Validation of Key Use Cases. In *Mile-High Video Conference (MHV '23), May 7–10, 2023, Denver, CO, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3588444.3591031

OVERVIEW

The Consumer Technology Association (CTA)'s Web Application Video Ecosystem (WAVE) project released CTA-5006 - Common Media Server Data (CMSD) [10] in Nov. 2022. This specification defines a standard means by which every media server (origin and intermediate) can communicate data with each media object response and have it received and processed consistently by every intermediary and media player to improve the efficiency and performance of distribution, and ultimately, the quality of experience (QoE) enjoyed by the users.

This paper presents an overview of CMSD implementations (media players and servers) and evaluations based on these implementations. From the media player, dash.js, the reference player of the DASH Industry Forum [12], has been updated to support CMSD by interpreting the two HTTP response headers defined in the specification: CMSD-Static and CMSD-Dynamic. This follows the already existing implementation of Common Media Client Data (CMCD) [11] in dash.js [5, 7, 14, 16, 18]. The integration of CMSD into dash.js enables communication from servers to media players and to evaluate different optimization opportunities [4].

The first use case we evaluate is leveraging edge server throughput estimates (etp) to improve the selection of the initial bitrate. A media player's rate-adaptation algorithm often conservatively selects lower bitrates at the startup due to the lack of throughput information. At the startup phase, an estimated throughput value can hint an appropriate starting bitrate level allowing the player to experience higher quality sooner. After the startup phase, the throughput estimates can be combined with player-side throughput estimations to make rate-adaptation decisions more accurate.

Server-side throughput estimation can also be used for the entire playback session to improve the rate-adaptation algorithms, specifically for low-latency streams based on chunked transfer encoding [6]. The bandwidth estimation at the server side uses the transport layer's congestion control layer, leading to more precise estimates. The results obtained in [1, 9] illustrate that the server-side estimated throughput is accurate with respect to the effective available shaped bandwidth, while the client-side throughput calculation may overestimate the available bandwidth.

The second use case we evaluate is prefetching of manifests, playlists and segments between the origin and edge server using the next object response (nor) key. This potentially reduces the media player's startup delay and improves its average quality.

The third use case we investigate is allowing an edge server to signal a suggested playback bitrate to improve the collective QoE. This is accomplished using the max suggested bitrate (mb) key. As a result, all media players receiving this information voluntarily reduce their total bitrate to the value indicated by this key. We confirm that an individual dash.js player honors the requested bitrate level. A cohort of competing dash.js players on a throughput-constrained network [2] also achieves a better aggregate QoE by complying with the (mb) instructions. Related research in this area are [3, 8, 13, 15, 17].

MHV '23, May 7–10, 2023, Denver, CO, USA

ACM ISBN 979-8-4007-0160-3/23/05.
https://doi.org/10.1145/3588444.3591031

REFERENCES

[1] Testing Low Latency: S4Streaming Against Client-Side Algorithms. [Online] Available: https://marketing.broadpeak.tv/l/310951/2022-07-27/34fqqc2/310951/1658924402tjch5li4/WhitePaper_Testing_Low_Latency_S4Streaming.pdf. Accessed on Apr. 1, 2023.

[2] 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).

[3] 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).

[4] A. C. Begen. Manus manum lavat: media clients and servers cooperating with common media client/server data. In *ACM Applied Networking Research Wksp. (ANRW)*, 2021 (DOI: 10.1145/3472305.3472886).

[5] A. C. Begen, A. Bentaleb, D. Silhavy, S. Pham, R. Zimmermann, and W. Law. Road to salvation: streaming clients and content delivery networks working together. *IEEE Commun. Mag.*, 59(11):123–128, 2021 (DOI: 10.1109/MCOM.121.2100137).

[6] A. Bentaleb, M. N. Akcay, M. Lim, A. C. Begen, and R. Zimmermann. Catching the moment with LoL+ in Twitch-like low-latency live streaming platforms. *IEEE Trans. Multimedia*, 24:2300–2314, 2022 (DOI: 10.1109/TMM.2021.3079288).

[7] 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).

[8] A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, and R. Zimmermann. Meta reinforcement learning for rate adaptation. In *IEEE INFOCOM*, 2023.

[9] G. Bichot and N. L. Scouarnec. Server-side segment selection for low-latency streaming—S4S. *SMPTE Motion Imaging Jour.*, 130(10):50–56, 2021.

[10] Consumer Technology Association. CTA-5006: Web application video ecosystem–common media server data, Nov. 2022.

[11] Consumer Technology Association. CTA-5004: Web application video ecosystem–common media client data, Sept. 2020.

[12] DASH-IF. DASH Reference Client. [Online] Available: https://reference.dashif.org/dash.js/. Accessed on Apr. 1, 2023.

[13] M. Lim, M. N. Akcay, A. Bentaleb, A. C. Begen, and R. Zimmermann. The benefits of server hinting when DASHing or HLSing. In *ACM MHV*, 2022 (DOI: 10.1145/3510450.3517317).

[14] NUS-OzU. CMCD-DASH. [Online] Available: https://github.com/NUStreaming/CMCD-DASH. Accessed on Apr. 1, 2023.

[15] NUS-OzU. CMSD-DASH. [Online] Available: https://github.com/NUStreaming/CMSD-DASH. Accessed on Apr. 1, 2023.

[16] S. Pham, M. Avelino, D. Silhavy, T.-S. An, and S. Arbanowski. Standards-based streaming analytics and its visualization. In *ACM MMSys*, 2021.

[17] S. Pham, P. Heeren, C. Schmidt, D. Silhavy, and S. Arbanowski. Evaluation of shared resource allocation using SAND for ABR streaming. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(2s), July 2020.

[18] D. Silhavy. dash.js - Common-Media-Client-Data. [Online] Available: https://websites.fraunhofer.de/video-dev/dash-js-common-media-client-data-cmcd/. Accessed on Apr. 1, 2023.