Multi-Regional, Multi-CDN Delivery Optimizations using HLS/DASH Content Steering Standard

Yuriy Reznik
Brightcove, Inc.
Boston, MA, USA
yreznik@brightcove.com

Guillem Cabrera
Brightcove UK, Ltd.
London, UK
gcabrera@brightcove.com

ABSTRACT

Content steering is a recently added feature in HLS and DASH streaming standards, simplifying the design of multi-CDN delivery systems. One natural use of this technology is multi-regional, QOE-driven delivery optimizations. However, the design of steering servers for this purpose is not a trivial task. This paper offers problem formulations, shows that such problems are tractable and solvable by the existing methods, and then discusses the design of a system, reducing the proposed methods to practice. It also presents the results of an experimental study confirming that the proposed method is effective and leads to significant improvements in real-world multi-regional streaming tests utilizing 3 major CDN vendors.

KEYWORDS

Content delivery, HLS, DASH, CDN, Content steering.

1 INTRODUCTION

Content steering [1-7] is a new mechanism supported by the HLS [8] and DASH [9] streaming standards, enabling the use of multiple CDNs for media delivery. Figure 1 illustrates the operational principles of this technology. In addition to the existing streaming system elements, such as origin servers, CDNs, and streaming clients, it introduces a new network entity – a content steering server. This server communicates with the clients directly and instructs them which CDNs to use. The TTL (time to live) parameter in each server response controls the frequency of server-client exchanges.

Among natural utilities of content steering technology are QOS- and QOE-based optimizations. For example, if the performance of any of the CDNs degrades, the server can be programmed to start moving traffic away from it. However, the design of steering servers is not trivial. For example, their effectiveness depends on the TTL parameter. The shorter it is, the faster the server can react and make switch decisions. However, reducing it, e.g., to 10 seconds, significantly increases the load on the steering server. It also increases the operating costs. Hence, one must figure out how to deploy and operate such servers inexpensively, at scale, and allowing short TTL in-

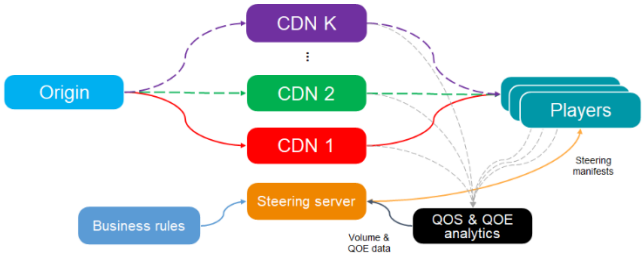


Figure 1: DASH streaming system with content steering.

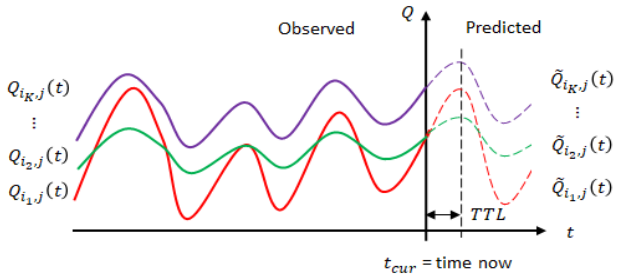


Figure 2. Conceptual view of QOE statistics for each CDN (i) in a region (j). TTL is an interval to the next decision.

teractions [10-13]. Additionally, one typically has to approach the QOE optimization problem holistically, accounting for both short-term and systematic differences in the performance of CDNs in different regions and limits imposed on the CDN traffic in the forms of global target load allocations or CDN commits [14]. All such factors are important considerations for practical multi-CDN system designs.

Considering such challenges, this paper offers a possible approach for formalizing and solving QOE-based optimization problems in the design of steering servers for multi-CDN streaming systems. Among related prior work, we must mention references [10-13], offering details about the content steering standard and its potential. References [12,13] offer ideas about scalable steering server design. References [11,12] advocate using in-session-level QOE-driven steering decisions. Reference [14] discusses the management of CDN commits and related cost-based optimizations. Reference [10] discusses cost-driven multi-CDN optimization problems with QOE-based

constraints. Relative to the reference [10], this paper presents an inverse and more precisely defined optimization problem. It also shows that this problem belongs to a class of linear programming problems, and hence, it is solvable by using the existing methods.

The rest of the paper is organized as follows. Section 2 defines the problem and shows that it is solvable by using the existing techniques. Section 3 describes the design of an experimental system, reducing it to practice. Section 4 discusses the obtained experimental results. Section 5 offers conclusions.

2 FORMALIZATION OF A PROBLEM

Consider a multi-CDN system with K CDNs and R operating regions. Such regions can be identified, e.g., by country codes, sub-divisions, ISP ASN numbers, etc.

As illustrated in Figure 2, let us also assume that for each CDN (i) and each region (j), we can observe chains of "quality scores" $Q_{ij}(t)$ reported over time (t):

$$Q_{ij}(t), \quad i = 1, \dots K, \;\; j = 1, \dots, R \tag{1}$$

Such scores may represent a specific QOS or QOE metric (e.g., buffer ratio) or a fused combination of several metrics (e.g., average bandwidth, latency, startup time, buffering time, resolution, the number of rendition switches, etc.) observed by the system. ITU-T Recommendation P.1203 [16] is a good example of a "fused" quality metric that may be employed for this application.

As also shown in Figure 2, for the decision-making, we will need not only quality scores observed in the past $t \le t_{cur}$, but also predicted scores for the duration of the next steering period $t \in (t_{cur}, t_{cur} + TTL]$. We assume that such predictions can be computed by using a combination of long-term (e.g., 1-year, 1-week, and 1-day intervals) as well as short-term (e.g., last minute) prediction samples and classic prediction techniques (such as linear prediction [17]).

Let us next assume that we have a matrix $\xi \in \Xi$:

$$\Xi = \left\{ \begin{bmatrix} \xi_{11} & \cdots & \xi_{1R} \\ \vdots & \ddots & \vdots \\ \xi_{K1} & \cdots & \xi_{KR} \end{bmatrix}, \; \forall i,j: \xi_{ij} \in [0,1], \; \sum_{i=1}^{K} \sum_{j=1}^{R} \xi_{ij} = 1 \right\} \tag{2}$$

defining load factors for all CDNs in all regions.

Next, by using this matrix, we can compute the average quality delivered by the system at time (t):

$$\bar{Q}(\xi, t) = \sum_{i=1}^{K} \sum_{j=1}^{R} \xi_{ij} Q_{ij}(t). \tag{3}$$

The average predicted quality in the next steering interval also becomes computable as:

$$\bar{Q}(\xi) = \frac{1}{TTL} \int_{t_{cur}}^{t_{cur}+TTL} \bar{Q}(\xi, t)\, dt. \tag{4}$$

Let us next consider global-level load factors on each CDN:

$$\zeta_i = \sum_{j=1}^{R} \xi_{ij}, \; \mathrm{i} = 1, \dots, \mathrm{K}. \tag{5}$$

In practical systems, CDN load factors (5) must be constrained to satisfy the commit conditions defined by the CDN contracts. For example, we may have a vector $\zeta_{min} = [\zeta_{min,1}, \dots, \zeta_{min,K}]$, prescribing minimum load levels for each CDN:

$$\zeta_i \ge \zeta_{min,i}, \quad i = 1, \dots K. \tag{6}$$

In some other cases, the multi-CDN systems may also operate with manually defined load balance factors:

$$\zeta_i = \zeta_{target,i}, \quad i = 1, \dots K. \tag{7}$$

Combining these requirements allows us to pose the following optimization problems. For the next steering period, find a matrix of load factors: $\xi^* \in \Xi$, such that:

$$\bar{Q}(\xi^*) = \min_{\substack{\xi \in \Xi \\ \sum_{j=1}^{L} \xi_{ij} \ge \zeta_{min,i}, i=1,\dots,K}} \bar{Q}(\xi). \tag{8}$$

or

$$\bar{Q}(\xi^*) = \min_{\substack{\xi \in \Xi \\ \sum_{j=1}^{L} \xi_{ij} = \zeta_{target,i}, i=1,\dots,K}} \bar{Q}(\xi). \tag{9}$$

As we immediately notice, both problems are variants of a classic linear programming problem [18]. The cost function (4) is a linear function of the matrix parameters ξ_{ij}, and there is a total of $K \times (R-1)$ unknown parameters within this matrix that we need to find. This problem can be easily solved by using existing methods [13].

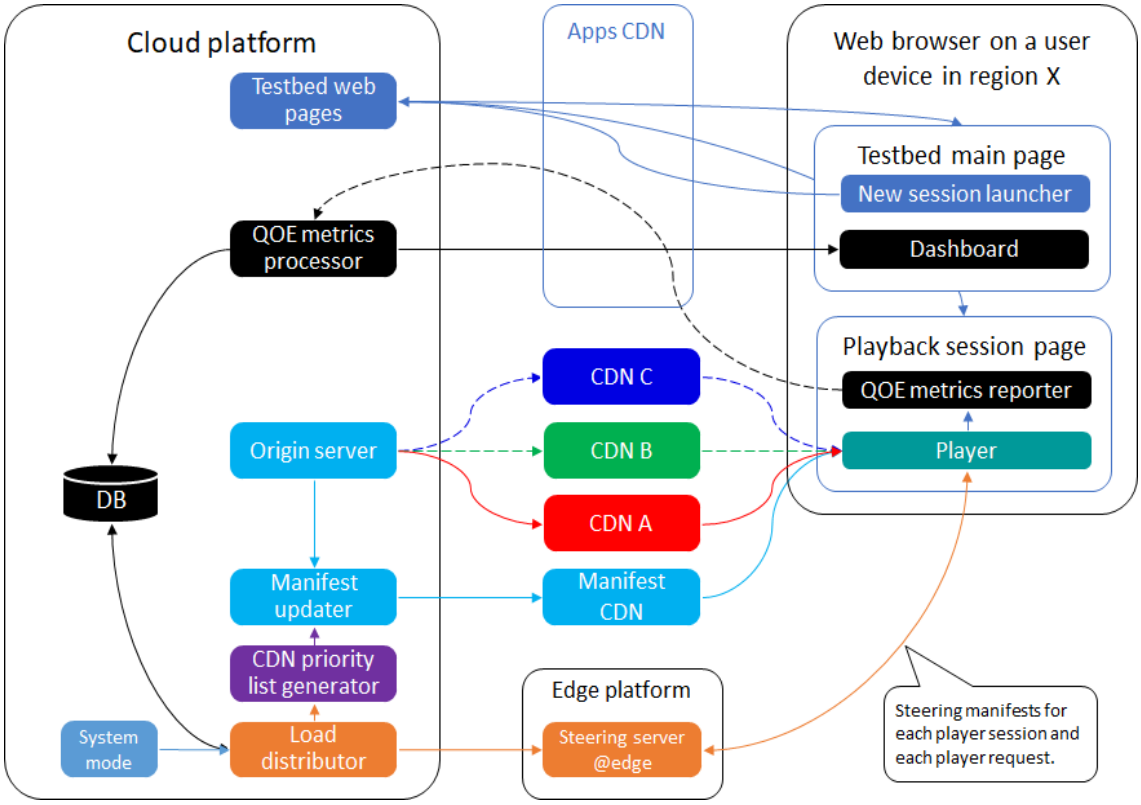


Figure 3. Experimental framework.

Practically, the above problems may have additional constraints, such as, for example, a constraint on the variation of load distribution assigned for the next period relative to the load distribution used in the past. However, such contrarians should not change the nature of the problems or affect their tractability.

Once the optimal load factors ξ_{ij} for each CDN and in all regions are computed, the system can use them to guide the subsequent steering decisions. The following section will offer additional details on such a system design.

3 IMPLEMENTATION AND VALIDATION

3.1 Experimental framework

Figure 3 shows the architecture of a system designed to validate the proposed method. It uses three tier-1 global-scale commercial CDNs, anonymized as CDN-A, CDN-B, and CDN-C. They all pull content from an origin (AWS S3 server, operating in NA). The system also employs a load distributor module that computes the matrix of load factors ξ according to the proposed method.

As input, the load distributor receives QOE and QOS statistics reported by players and processed by a data pipeline instrumented as part of the system. Such statistics are collected

Table 1: Volume, QOS, amd QOE metrics

Category	Metric description	Unit
Volume	Video views	#
	Seconds played	seconds
	Traffic volume	GB
QOS	Average throughput	Mbps
	Standard deviation of throughput	Mbps
	Average latency	ms
	Standard deviation of latency	ms
QOE	Startup time	ms
	Buffer ratio (buffering/play time)	%
	Buffering events	#/session
	Video bitrate	Mbps
	Video resolution (height)	lines
	Rendition switches	#/session

Table 2: Parameters of encoded HLS/DASH streams.

Type	Codec	Profile	Bitrate	Resolution	Framerate
Video	H.264	High	4531	1920x1080	30
Video	H.264	High	2445	1280x720	30
Video	H.264	Main	1419	1024x576	30
Video	H.264	Main	783	640x360	30
Audio	AAC	LC	128		

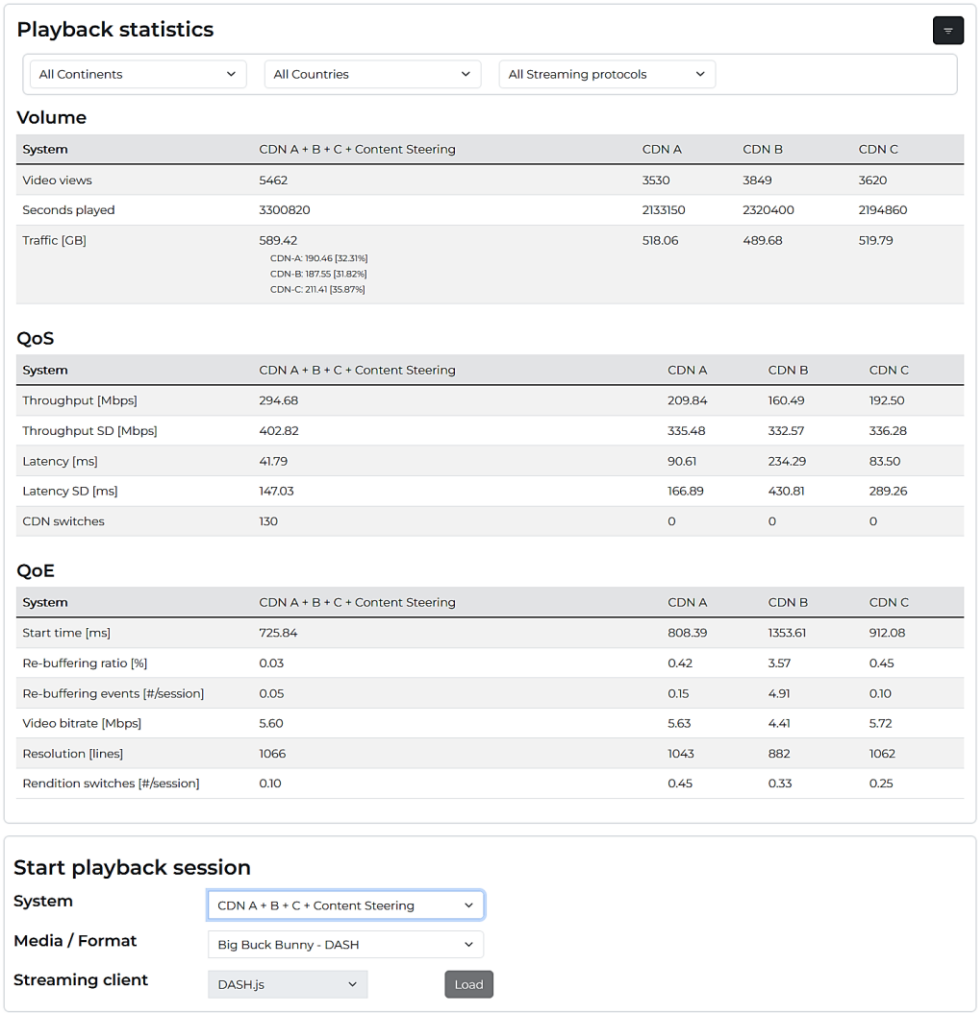


Figure 4. Main page of the testing framework. It shows overall statistics and allows users to start new playback sessions.

and aggregated for each continent and country as regions. Table 1 lists the metrics used by this system. The combined "quality score" metric used for optimizations was:

$$Q = \frac{\text{video resolution}}{(1 + \text{buffering events}) \cdot (3 + \text{rendition switches})}. \quad (10)$$

This metric was chosen mainly for its conceptual simplicity. It grows linearly with the resolution of delivered video and decreases inverse proportionally to the frequency of buffering events and rendition switches.

The load distributor pulls relevant QOE statistics and computes the effective matrix ξ every 10 minutes. It uses our problem formulation (9) and simplex method to solve the optimization problem.

The CDN priority list generator computes the initial CND assignments for each playback session. It is essentially a random number generator shaped by the probability of CDNs implied by the matrix ξ in the session's region. The manifest

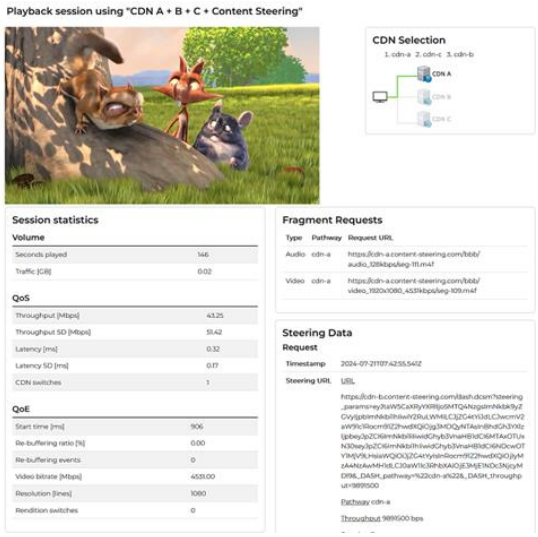
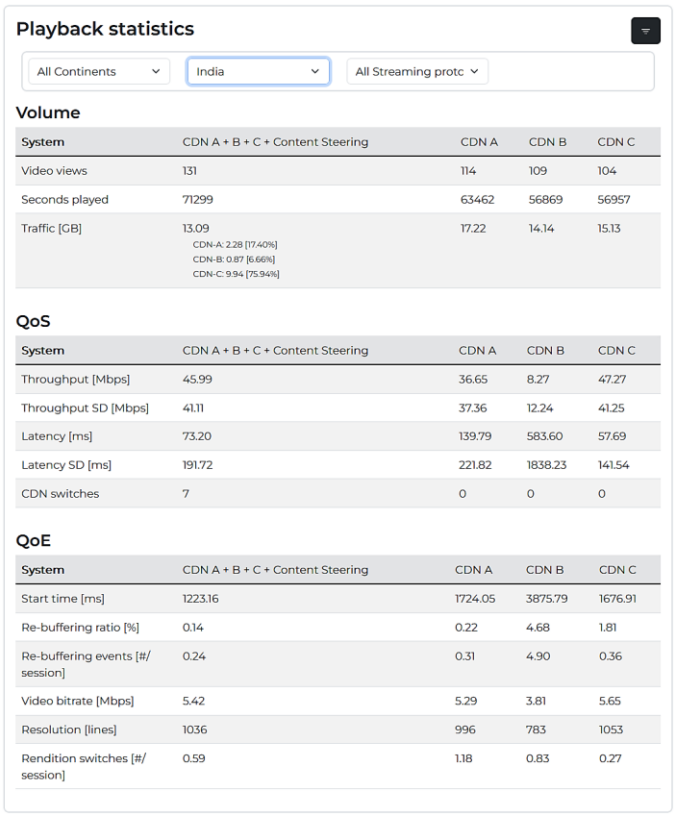


Figure 5. The page with a playback session in progress



Playback statistics

All Continents | India | All Streaming protc

Volume

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Video views	131	114	109	104
Seconds played	71299	63462	56869	56957
Traffic [GB]	13.09 CDN-A: 2.28 [17.40%] CDN-B: 0.87 [6.66%] CDN-C: 9.94 [75.94%]	17.22	14.14	15.13

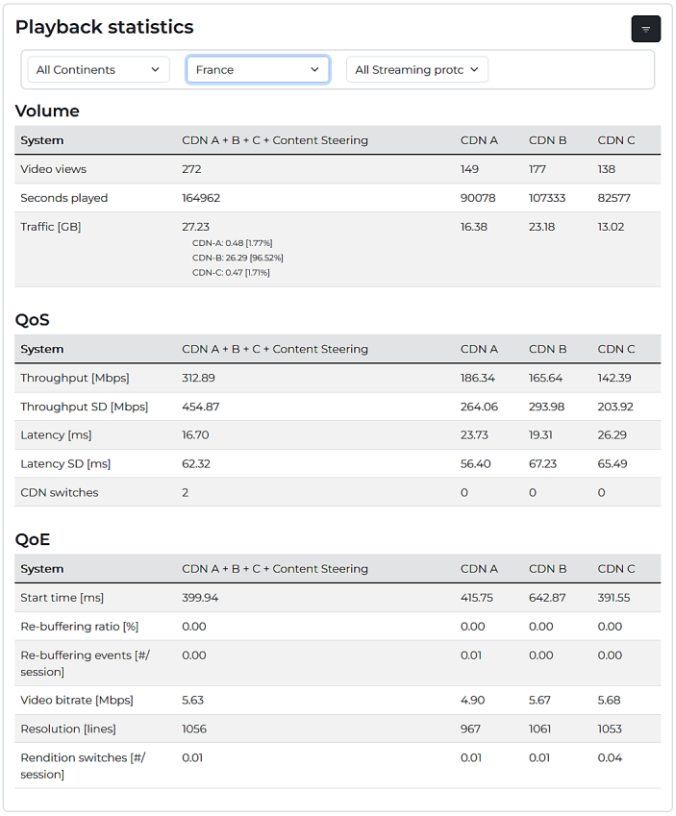
QoS

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Throughput [Mbps]	45.99	36.65	8.27	47.27
Throughput SD [Mbps]	41.11	37.36	12.24	41.25
Latency [ms]	73.20	139.79	583.60	57.69
Latency SD [ms]	191.72	221.82	1838.23	141.54
CDN switches	7	0	0	0

QoE

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Start time [ms]	1223.16	1724.05	3875.79	1676.91
Re-buffering ratio [%]	0.14	0.22	4.68	1.81
Re-buffering events [#/session]	0.24	0.31	4.90	0.36
Video bitrate [Mbps]	5.42	5.29	3.81	5.65
Resolution [lines]	1036	996	783	1053
Rendition switches [#/session]	0.59	1.18	0.83	0.27

Figure 6. Playback statistics collected in India.



Playback statistics

All Continents | France | All Streaming protc

Volume

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Video views	272	149	177	138
Seconds played	164962	90078	107333	82577
Traffic [GB]	27.23 CDN-A: 0.48 [1.77%] CDN-B: 26.29 [96.52%] CDN-C: 0.47 [1.71%]	16.38	23.18	13.02

QoS

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Throughput [Mbps]	312.89	186.34	165.64	142.39
Throughput SD [Mbps]	454.87	264.06	293.98	203.92
Latency [ms]	16.70	23.73	19.31	26.29
Latency SD [ms]	62.32	56.40	67.23	65.49
CDN switches	2	0	0	0

QoE

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Start time [ms]	399.94	415.75	642.87	391.55
Re-buffering ratio [%]	0.00	0.00	0.00	0.00
Re-buffering events [#/session]	0.00	0.01	0.00	0.00
Video bitrate [Mbps]	5.63	4.90	5.67	5.68
Resolution [lines]	1056	967	1061	1053
Rendition switches [#/session]	0.01	0.01	0.01	0.04

Figure 8. Playback statistics collected in France.

updater then embeds a list of the CDNs in the manifest and passes it to the steering server, which is responsible for all subsequent CDN switch decisions.

Our system uses Fastly's Compute@Edge platform to deploy content steering servers. Such edge-based deployment scales well and allows frequent client-server exchanges [12]. The edge servers receive the initial order of CDNs and then make switch decisions during each session. For example, they may move the current CDN to the last position in the priority list if the client reports serious problems (e.g., insufficient throughput or network errors [12]). It works well as failover and error reduction logic. However, such events are rare. Most commonly, the edge servers preserve the CDN order assigned by the load distributor. The TTL of steering server-client exchanges in our implementation is 10 sec.

As test video content, this system uses the classic 10-minute "Big Buck Bunny" sequence [22]. Both HLS and DASH streams form ladders, as shown in Table 2.

As a DASH player, the system uses the DASH.js player [20]. As an HLS player, it employs HLS.js [21].

All tests were executed by orchestrating playback sessions on servers in different world regions (our test set of regions included 21 countries in six continents).

3.2 System operation

Figure 4 shows the main web page of the system. It includes the playback statistics dashboard and a tool for launching new sessions. The "playback statistics" panel shows the metrics collected for the following four operating modes of the system:

- CDN A + CDN B + CDN C + content steering
- CDN A
- CDN B
- CDN C

This combination of modes allows users to see how a multi-CDN system with steering compares against the performance achievable by any single CDN. Users can see such statistics for each continent, country, and streaming protocol.

The "start playback session" section allows users to start new sessions. Entering configuration and clicking the "load" button brings a new page with a web player, CDN selection window, and session-level statistics, as shown in Figure 5.

3.3 Experimental results

Figure 4 shows the overall statistics collected by this system in our experiments. It reports over 5000 sessions executed using a system with content steering and over 16000 sessions

overall. The overall playback time delivered by all systems is over 2600 hours, and the overall volume of media data delivered is over 2000 GB.

The current global traffic distribution between the CDNs in a system with content steering is [32.31%, 31.82%, 35.87%]. It is close to an even split $\xi_{target} = [1/3,1/3,1/3]$, provided as a target to the load distributor. However, we note that the resulting traffic distributions between CDSs in each region are different. This effect is the consequence of our optimization algorithm's work. For instance, Figure 6 and Figure 7 show the results for India and France, respectively. In India, we see that CDN-B is the worst; hence, it receives only 6% of traffic, while in France, it is about as good as any other CDN, and here it gets almost all traffic. We also note different numbers for in-session CDN switches reported by our system. For instance, in India, where CNDs struggle more, the edge servers executed seven such switches, while in France, only 2.

The net effects of such traffic allocation and switch decisions are (a) significantly better QOE achieved in regions with poor CDN performance and (b) better QOE achieved overall. Based on statistics in Figure 4, we note that the average buffering ratio for the 3-CDN system with steering is only 0.03%, while for the best single CDN system, it jumps to 0.42%. The frequency of buffering events per session has also decreased to 0.05 events/session vs. 0.1 for the best-performing single CDN system. We also note some improvements in the average resolution of videos delivered: 1066 lines vs. 1062 lines, and in reducing the number of rendition switches: 0.10 vs. 0.25 for the best-performing single CDN system. In other words, we observe that our proposed multi-CDN streaming system works and significantly outperforms single-CDN systems using the same CDNs.

4 CONCLUSIONS AND FUTURE WORK

The paper presented a formalization of a multi-regional, multi-CDN delivery optimization problem with average system QOE used as an optimization criterion. We showed that this problem maps to the class of linear programming problems, and hence, it is solvable using the existing methods. We have also described an experimental system reducing the proposed method to practice. We also presented an experimental study confirming that this proposed method works and enables multi-CDN systems to achieve significant QOE improvements relative to single-CDN systems.

While the results are promising, we must also point out that our proposed framework is intentionally very simplistic. A more complete (and practically relevant) definition of the problem should also consider:

- deeper regionalization by also considering last mile and transit ISPs
- the effects of the popularity of the content
- the effects of CDN cache misses
- the delivery costs of the CDNs [14]
- the costs of origin servers and origin to CDN traffic
- quality-costs tradeoffs implied by business objectives

and other additional criteria and constraints. We plan to address some combinations of such additional factors in our future research.

REFERENCES

[1] Apple. 2021. HLS Content Steering Specification v1.2b1. (April 2021), https://developer.apple.com/streaming/HLSContentSteeringSpecification.pdf

[2] R. Pantos. 2021. RFC 8216bis: HTTP Live Streaming 2nd Edition, Section 7: Content steering. IETF (November 2021). https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis#section-7

[3] DASH-IF. 2022. DASH-IF Candidate Technical Specification: Content Steering for DASH, Version 0.9.0. DASH-IF (July 2022). https://dashif.org/docs/DASH-IF-CTS-00XX-Content-Steering-Community-Review.pdf

[4] ETSI. 2024. ETSI TS 103 998: Content Steering for DASH, v1.1.1, ETSI (January 2024). https://www.etsi.org/deliver/etsi_ts/103900_103999/103998/01.01.01_60/ts_103998v010101p.pdf

[5] Y. Reznik, G. Cabrera, D. Silhavy, S. Pham, A. Giladi, A. Balk, A. C. Begen, and W. Law. 2024. Content Steering: a Standard for Multi-CDN Streaming. In Proc. ACM Mile-High Video Conference (MHV '24) Denver, CO (February 11-14, 2024).

[6] Y. Reznik, G. Cabrera, D. Silhavy, S. Pham, A. Giladi, A. Balk, A. C. Begen, and W. Law. 2024. Content Steering: a Standard for Multi-CDN Streaming. In Proc. International Broadcast Convention (IBC 2024), Amstedam, NL (September 14-16, 2024).

[7] Y. Reznik, G. Cabrera, D. Silhavy, S. Pham, A. Giladi, A. Balk, A. C. Begen, and W. Law. 2025. Content Steering: a Standard for Multi-CDN Streaming. SMPTE Motion Imaging Journal (January/Febryary, 2025), 32-41.

[8] R. Pantos and W. May. 2017. RFC 8216: HTTP live streaming, IETF (2017). https://tools.ietf.org/html/rfc8216

[9] ISO/IEC. 2022. 23009-1:2022: Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats. ISO/IEC (October 2022).

[10] H. H. Liu, Y. Wang, Y. R. Yang, H. Wang, and C. Tian. 2012. Optimizing cost and performance for content multihoming. In Proc. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM). (2012) 371-382.

[11] D. Silhavy, S. Pham, A. Giladi, A. Balk, A. Begen, and W. Law. 2023. Real-Time Streaming Reliability and Performance Optimization Using Content Steering, In Proc. SMPTE Media Technology Summit, Hollywood, CA (October 16-19, 2023).

[12] Y. Reznik, G. Cabrera, R. Zekarias, B. Zhang, B. Panigrahi, N. Barman, S. Hicks, T. Krofssik, A. Sinclair, and A. Waldron. 2023. Implementing HLS/DASH Content Steering at Scale. In Proc. International Broadcast Convention (IBC 2023), Amstedam, NL (September 15-18, 2023).

[13] E. Gama, R. Rodrigues-Filho, E. Madeira, R. Immich, and L. F. Bittencourt. 2024. Enabling Adaptive Video Streaming via Content Steering on the Edge-Cloud Continuum, In Proc. IEEE International Conference on Fog and Edge Computing (ICFEC 2024), Philadelphia, PA, USA, May 06-09, 2024.

[14] Y. Reznik and G. Cabrera. 2024. On Multi-CDN Delivery Costs Optimization Problem, In Proc. IEEE International Symposium on Multimedia (ISM 2024), Tokyo, Japan, (December 8-11-13, 2024).

[15] CTA, CTA-5004: Common Media Client Data (CMCD). CTA (September 2020). https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5004-final.pdf

[16] ITU-T, P.1203: Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport, ITU-T (October 29, 2017). https://www.itu.int/rec/T-REC-P.1203-201710-I/en

[17] Gersho and R. Gray. 1992. Vector quantization and signal compression. Chapter 4: Linear Prediction. Kluwer, Boston, MA (1992) 83-125.

[18] D. Goldfarb and M. J. Todd. 1989. Chapter II Linear Programming. Handbooks in operations research and management science, Volume 1, Elsevier, NL (1989) 73-170.

[19] SVTA, 2024. Content Steering at Edge. https://github.com/streaming-video-technology-alliance/content_steering_at_edge (available to SVTA members)

[20] DASH-IF, Reference client, online: https://dashif.org/tools/dashjs/

[21] Hls.js player, online: https://github.com/video-dev/hls.js/

[22] Blender Foundation, Big Buck Bunny video sequence, https://peach.blender.org/