TECHNICAL PAPER

KEYWORDS STREAMING // DASH // HLS // CDN SWITCHING // CONTENT STEERING

Content Steering: A Standard for Multi-CDN Streaming

By Y. Reznik, G. Cabrera, D. Silhavy, S Pham, A. Giladi, A. Balk, A. Begen, and W. Law

Given CDN limits, large streaming operators increasingly employ multiple CDNs and so-called "CDN switching" technologies as part of their delivery architectures. By intelligently distributing traffic across multiple CDNs, such systems can achieve better reliability, scale, and quality of experience (QoE) delivered to end users.

1545-0279/25©2025SMPTE

Abstract
Content Steering for DASH (ETSI TS 103 998) is a new standard developed by the DASH Industry Forum (DASH-IF), defining means for managing media delivery using multiple content delivery networks (CDNs). At the server-client interaction level, this standard is compatible with the CDN steering features of HLS (IETF RFC 8216bis), effectively enabling the same content steering servers to control delivery for both HLS and DASH distributions. This paper reviews the history of this standard's creation, explains its operation principles, and discusses its various features, utilities, and benefits. The paper also surveys the available implementations of streaming clients and servers supporting this standard and the ongoing efforts in DASH-IF and Streaming Video Technology Alliance (SVTA) organizations to support the rollout of this technology in the industry. Finally, the paper presents the results of an experimental study of multi-CDN delivery systems conducted by SVTA. These results show significant QoE improvements achieved by a system using content steering.

As well known, most videos sent over the internet are delivered using streaming technologies.[1-6] The two most commonly used streaming protocols today are HTTP Live Streaming (HLS)[7] and Dynamic Adaptive Streaming over HTTP (DASH).[8] Both are international standards. Both use HTTP as the underlying network protocol and employ content delivery networks (CDNs) for distribution.[9,10]

The fundamental principle of HTTP-based streaming is simple: the media content is encoded, packaged, and placed on the origin server first. CDN then propagates it, through its system of caches, to a vast and geographically dispersed population of viewers. Effectively, CDN enables mass-scale delivery.[4,10]

However, CDNs have some limits. Some may not be available in all relevant regions. Some may have a saturated internal network, and some may not have sufficient capacity of edge caches to support the delivery of a vast and diverse catalog of media content. Occasionally, CDNs may also experience outages or technical failures, making them unavailable for some time.[9,10]

Given such limits, large streaming operators increasingly employ multiple CDNs and so-called "CDN switching" technologies as part of their delivery architectures.[10-12] By intelligently distributing traffic across multiple CDNs intelligently, such systems can achieve better reliability, scale, and quality of experience (QoE) delivered to end users. However, developing and operating such multi-CDN systems are not trivial tasks.[10,13,14]

Even a basic traffic switching operation between different CDNs is not exactly straightforward. **Table 1** lists several methods and approaches tried in the past. As easily observed, none of these approaches is perfect. Each has various cons and pros.[10-13]

Fortunately, this problem has been recently addressed by standards.[15-21] The latest versions of HLS and DASH specifications[15-17] now include "content steering" functions designed specifically for this purpose. Using these functions, the design of multi-CDN delivery systems becomes much more straightforward. Much less effort and fewer changes are required across the streaming workflow. It also ensures proper and consistent switching behavior for all players implemented according to HLS and DASH standards. Once deployed, it is also guaranteed to work with subsequent system component upgrades (players, packagers, etc.). The design becomes simple, reliable, and future-proof.

This paper reviews the content steering technology and discusses its present state of implementation, validation, and adoption by the industry. Section 2 reviews the history of this standard's development, its principles of operation, and key features offered. Section 3 surveys the available implementations of streaming clients, servers, and other tools supporting this standard. Section 4 brings recent validation and performance test results. Section 5 drives conclusions.

The Standard

Standard Development

The initial concept of content-steering technology for HLS was developed by Apple in February 2021[22]. It was added to IETF RFC 8216bis, "HTTP Live Streaming 2nd Edition," in November 2021.[15]

Subsequently, in July 2022, the DASH Industry Forum (DASH-IF) produced a technology proposal titled "Content Steering for DASH."[23] The DASH-IF proposal has extended the HLS content steering concept by preserving the syntax of the client-server exchanges and defining a few new ele-

1545-0279/25©2025SMPTE

ments specific to DASH. This document was published for community review and has received many comments from the engineering community. The updated text of the DASH-IF content steering specification, addressing all these comments, was produced in October 2023. Subsequently, it was submitted for standardization to the European Telecommunications Standards Institute (ETSI) and published as ETSI standard TS 103 998[17] in February 2024. In parallel, the corresponding changes to the DASH MPD syntax have also been submitted to MPEG and incorporated into the 6th edition of MPEG-DASH standard ISO/IEC DIS 23009-1:2024.[16]

Principles of Operation

To illustrate the main principles of the content steering mechanism, in **Fig. 1**, we depict an example of a streaming delivery system practicing it. This system employs two CDNs delivering encoded media data and another CDN delivering the manifests (MPD files for DASH or master playlists for HLS). The service locations (or "pathways") of media CDNs are denoted as "alpha" and "beta," respectively. The system also deploys a new server-side element—the content steering server.

Service locations of CDNs and steering servers deployed by the system, as well as their initial/default CDN to be used for delivery, are defined in the manifests. In DASH manifest files, the corresponding syntax elements are BaseURLs and a ContentSteering descriptor:

```
<BaseURL serviceLocation="alpha">https://cdn1.com/</BaseURL>

<BaseURL serviceLocation="beta">https://cdn2.com/</BaseURL>

<ContentSteering defaultServiceLocation="beta" queryBefore-
Start="true">https://steeringserver.com

</ContentSteering>
```

In HLS, the corresponding syntax elements include redundant variant streams with PATHWAY-ID annotations and an #EXT-X-CONTENT-STEERING tag:

```
#EXTM3U

#EXT-X-CONTENT-STEERING:SERVER-URI="https://steeringserver.com",PATH-
WAY-ID="beta"

#EXT-X-STREAM-INF:BANDWIDTH=1280000,PATHWAY-ID="alpha"

https://cdn1.com/hi/video.m3u8

#EXT-X-STREAM-INF:BANDWIDTH=1280000,PATHWAY-ID="beta"

https://cdn2.com/hi/video.m3u8
```

When HLS or DASH streaming clients receive such manifests, they recognize the presence of the steering servers and start

TABLE 1. Comparison of the Existing Methods for CDN Switching.[12,13]

Method	Pros	Cons
DNS-based	It is the simplest of all solutions since the source video URL remains constant.	The switch delay is more time-consuming, ranging from 300 sec up to 15 min in the event of CDN failures.. This can immensely hamper the user QoE.
Manifest rewrite	Enables midstream switching for live streams. No matter the volume of simultaneous session resets, this method reduces the chances of a cascade effect that may hamper the video workflow.	Rewriting the manifests can sometimes bring about errors. Midstream switching is not entirely seamless, as it takes time for the server to understand that a particular CDN is unavailable.
Server-side	It is a relatively simple CDN switching method to implement and deploy since the server makes all the changes. It is also easier for the operator to control.	Page loading may take some time, adding to delays. Since CDN switching is based on the collective data from many clients, it does not necessarily consider the unique conditions of the actual clients.
Client-side	Quality of Service (QoS) data is almost accurate as it is fetched based on individual clients' local and real-time performance metrics. Seamless midstream CDN switching is possible.	It is a complex procedure to implement when built in-house due to the code complexity of the algorithms, which requires detailed planning. It may not be feasible for platforms with "closed" players.

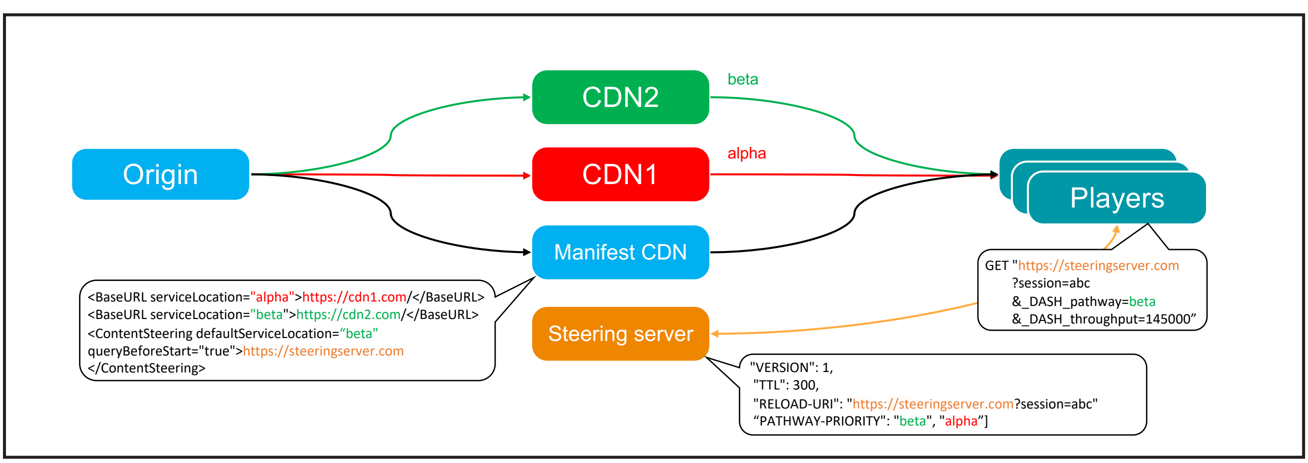


FIGURE 1. Multi-CDN delivery system with content steering.

1545-0279/25©2025SMPTE

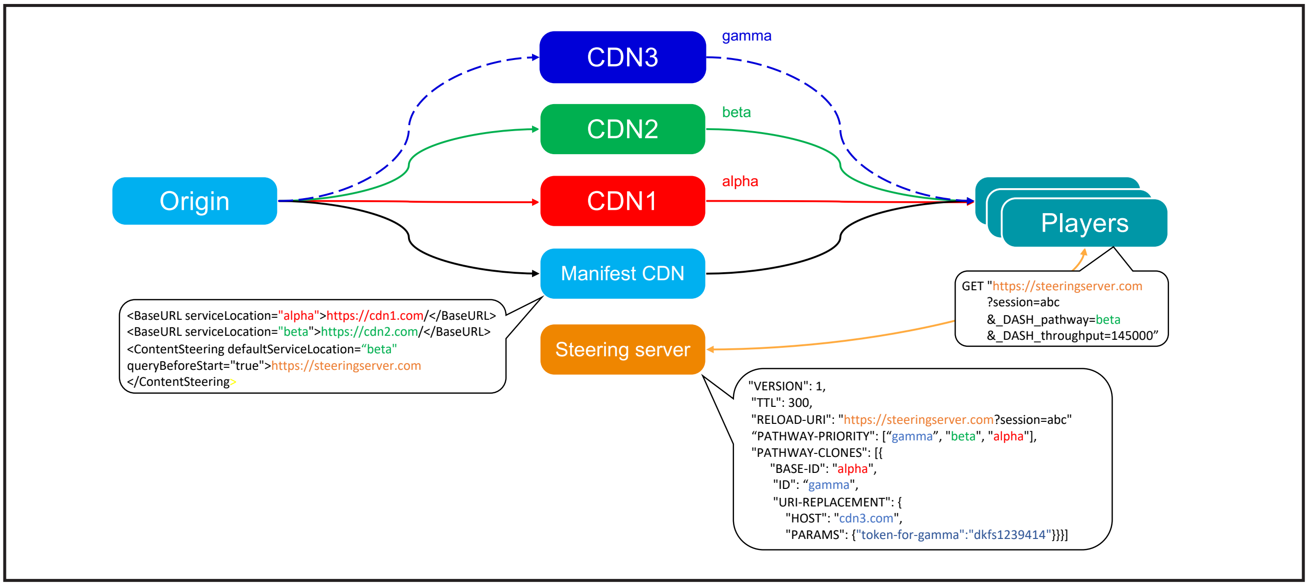


FIGURE 2. Dynamic addition of a third CDN in the streaming system using pathway cloning.

calling them by issuing HTTP GET requests to server URIs as specified in the manifests. As part of such requests, the clients may include some parameters. For example, a DASH client may issue the following request:

```
GET "https://steeringserver.com?session=abc&_DASH_pathway_=beta&_DASH_through-
put_=789320"
```

In this example, the client sends a session ID, the pathway ID, and the measured throughput parameters. **Table 2** lists standard parameters defined by HLS and DASH content steering specifications that clients may use. However, clients may also pass additional parameters, including, for example, CMCD metadata.[24,25]

In response to receiving a request, the content streaming server generates a response indicating the preferred order of the CDNs (or pathways), the time to call the steering server again (TTL), and the URI to use next time when calling the server. For example, the server may produce the following response (steering manifest):

```
{
 "VERSION": 1,
 "TTL": 300,
 "RELOAD-URI": "https://steeringserver.com?session=abc"
 "PATHWAY-PRIORITY": ["beta", "alpha"]
}
```

In this example, the server instructs the client to use pathway "beta" with a higher priority for streaming and then to call the server back in 300 sec for the next update. The 300 sec (5 min) TTL is a default response interval recommended by HLS specifications.

Once the client receives such a response, it checks if the top CDN in the priority list matches the one currently being used, and if not, it performs the switch.

Pathway Cloning

In addition to defining priorities for the CDNs listed in the manifest at the beginning of the streaming session, the content steering servers may also add new CDNs dynamically. This mechanism is called "pathway cloning." We illustrate its operation in **Fig. 2.**

This system is almost identical to the one we discussed earlier (see **Fig. 1**). However, in addition to the first two media CDNs, it now introduces the third, denoted as pathway "gamma." This third CDN is absent in the original manifest. Instead, the content steering server introduces it dynamically by sending the following instructions to the client.

```
{
 "VERSION": 1,
 "TTL": 300,
 "RELOAD-URI": "https://steeringserver.com?session=abc"
 "PATHWAY-PRIORITY": ["gamma", "beta", "alpha"],
 "PATHWAY-CLONES": [{
          "ID": "gamma",
          "BASE-ID": "alpha",
          "URI-REPLACEMENT": {
                "HOST": "cdn3.com",
                "PARAMS": {"token-for-gamma":"kdr1239414"}
         }
 }]
}
```

This steering manifest tells the client that the highest priority CDN is now "gamma," representing a new service location built by cloning. To synthesize it, the player would parse the PATHWAY-CLONES array to locate the definition of "gamma." It would then construct a pathway "gamma" by taking the URI for pathway "alpha" as a template, substituting the HOST component of the URL with "cdn3.com," and append-

TABLE 2. Query Parameters Defined for Client-server Exchanges.

HLS parameter	DASH parameter	Description
_HLS_pathway_	_DASH_pathway_	ID of the last pathway used by the client
_HLS_throughput_	_DASH_throughput_	Throughput [bits/sec], as observed by the client in pulling data from the selected CDN

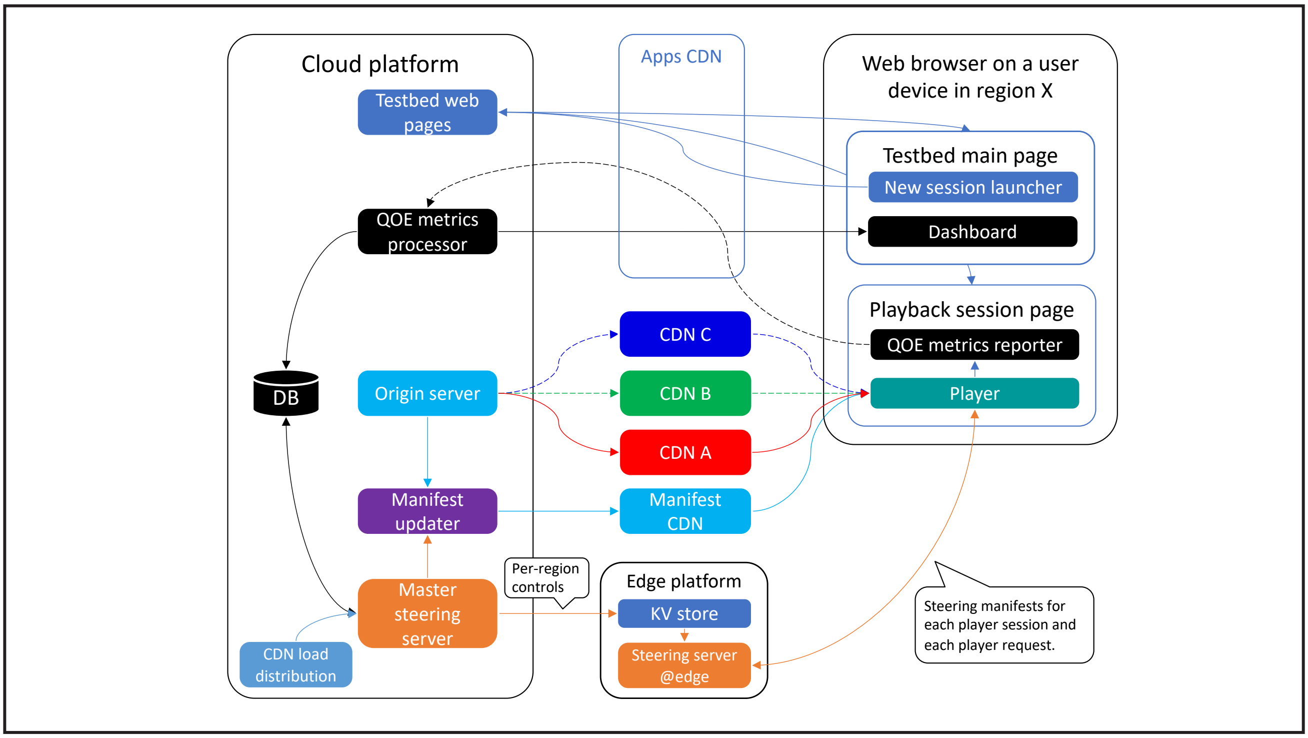


FIGURE 3. SVTA multi-CDN testbed with edge-deployed content steering servers.

ing the query argument with the "token-for-gamma" string defined in the steering response.

The pathway cloning mechanism is convenient for systems that do not perform manifest updates, enabling the content generated once to be delivered by different CDNs in the future. It may also be helpful in systems with the dynamic discovery of local caches, such as the SVTA Open Caching initiative.[26]

Client Behavior

Content steering specifications for HLS[15] and DASH[16,17] define how streaming clients should respond to steering manifests. They state, for instance, that streaming clients must always follow the priority order specified in steering manifests. However, they also allow clients to make variant-stream-level decisions while switching from one CDN to another. Such stream-level adaptation may help preserve the continuity of the playback. The DASH clients may also switch between alternative adaptation sets if multiple adaptation sets are available. With all these measures, the clients have the tools to execute CDN traffic switching seamlessly (or at least most gracefully, given the CDN and network conditions). The capability to execute seamless mid-session switches is one of the critical advantages of this standard.

The mechanism defined by the standard is highly robust. Offering clients a list of several CDNs instead of one enables them to fall back to the next CDN in the priority list if the top CDN becomes unavailable (e.g. if attempts to retrieve streams result in network errors).

The robustness of the design also extends to interactions with the content steering servers. For example, if steering servers become unavailable (e.g., the client receives 410 error codes), the client is instructed to continue playback using the default CDN choice. In other words, even if steering servers will fail, they won't cause the failure of the delivery system.

Implementations

In parallel with work on the standardization of the content steering technology, DASH-IF was also updating its reference client (dash.js),[27] content preparation, and validation tools.[28,29] Concurrently, SVTA has also begun developing and testing content steering servers.[30-32] Many additional organizations and open-source communities have followed, producing many technologies and products with built-in support for this standard. This section reviews some of these technologies and products.

Streaming Clients

The list of HLS and DASH clients supporting content steering technology includes:

1545-0279/25©2025SMPTE

- dash.js, version 4.5.0 and later (DASH)[27]
- Apple AVplayer, since iOS version 15 (HLS)[33]
- HLS.js, since version 1.4.0 (HLS)[35]
- Video.js, since version 8.8.8 (HLS and DASH)[36]
- Shaka player, since version 4.6.0 (HLS and DASH)[37]
- Brightcove web player, since version 7.15.0 (HLS and DASH)[38]
- Bitmovin web player, since version 8.11.0 (HLS and DASH)[39]
- Radiant media player, since version 9.13.0 (HLS and DASH)[40]

In addition, work is currently underway to add support to ExoPlayer[41] video player in Android OS. The initial prototype of such an implementation was reported in Ref. 42 while the progress on adding full support to the product can be tracked in Ref. 43.

Packagers, Manifest Updaters, and Steering Servers

The list of existing media packages, manifest updaters, and server-side tools with support for the content steering standard includes:

- Apple HTTP Live Streaming (HLS) Tools[34]
- Shaka Packager[40] with an update developed by DASH-IF,[45]
- Comcast Content Steering Server prototype (SVTA open-source project)[31]
- Brightcove Content Steering @edge (SVTA open-source project)[32]
- EINBLIQ.IO Content Steering Server[46]

Reference Streams, Testbeds, and Validation Tools

Finally, we must mention several publicly available demos, reference streams, and tools developed for testing content steering technology. These include:

- DASH-IF Validations tools[28]
- DASH-IF demo and reference streams for content steering[29]
- Apple HTTP Live Streaming (HLS) Tools[34]
- Brightcove Content Steering@Edge demonstration
- SVTA content steering testbed[48]

Performance Study

This section reports the preliminary results of a study on the performance of content-steering technology currently under progression in SVTA.[26] This study uses the testbed.[48]

Testbed Architecture

Figure 3 presents the testbed's architecture. This system uses three tier-1 commercial CDNs, which are anonymized and called CDN-A, CDN-B, and CDN-C, respectively. It also uses content steering servers developed and contributed as an SVTA-labs project by Brightcove.[32,49]

The steering server implementation includes a master steering server operating in a cloud platform (AWS) and lightweight edge servers, executed per each steering request by an edge platform (Akamai Edge Workers or Fastly Compute@Edge). The master steering server defines and controls the load allocation for CDNs in all regions the system supports. It also generates the initial priority lists of CDNs for each session. It passes such lists as query string parameters to the edge servers. The edge servers make all subsequent decisions. For example, if the performance of the top CDN is sufficient, the edge server will maintain the same CDN order throughout the session. However, if the client signals that the top CDN lacks throughput, the edge server may adjust the priority order to move traffic to a better-performing CDN. Ref. 49 offers additional details about the design of steering servers in this testbed.

TABLE 3. Volume, QoS, and QoE Metrics Reported by the Testbed.

Metric category	Metric description	Units
Volume	Video views (number of sessions)	count
	Seconds played	seconds
	Traffic (amount of data pulled by the players)	Gbytes
QoS	Average throughput of the CDN-client connection	Mbits/s
	The standard deviation of throughput	Mbits/s
	Average latency of CDN-client connection	ms
	The standard deviation of latency	ms
QoE	Startup time	ms
	Re-buffering ratio (buffering time/content duration)	%
	Re-buffering events	count/session
	Video bitrate	Mbits/s
	Video resolution (height)	lines
	Rendition switches	count/session

TABLE 4. Characteristics of the Encoded HLS/DASH Streams in the Testbed.

Media Type	Codec	Profile	Bitrate [kbps]	Resolution	Framerate
Video	H.264	High	4531	1920 x 1080	30
Video	H.264	High	2445	1280 x 720	30
Video	H.264	Main	1419	1024 x 576	30
Video	H.264	Main	783	640 x 360	30
Audio	AAC	AAC	128		

As input information, the master steering server uses QoE data collected by the QoE metrics processing engine. It is a basic QoE analytics system instrumented as part of the testbed. It receives periodic events from HLS and DASH clients playing the encoded test content. **Table 3** summarizes the metrics this system collects and reports.

Another input to the master steering server is a target load distribution that must be achieved across the CDNs on a global scale. By default, it is set to a uniform distribution, but it can be programmed. In extreme cases, the server can be instructed to route all traffic to a single CDN, effectively reducing the system to a single-CDN setup.

As test video content, the testbed employs the classic 10-min "Big Buck Bunny" sequence.[50] Both HLS and DASH streams follow the same encoding ladder, presented in

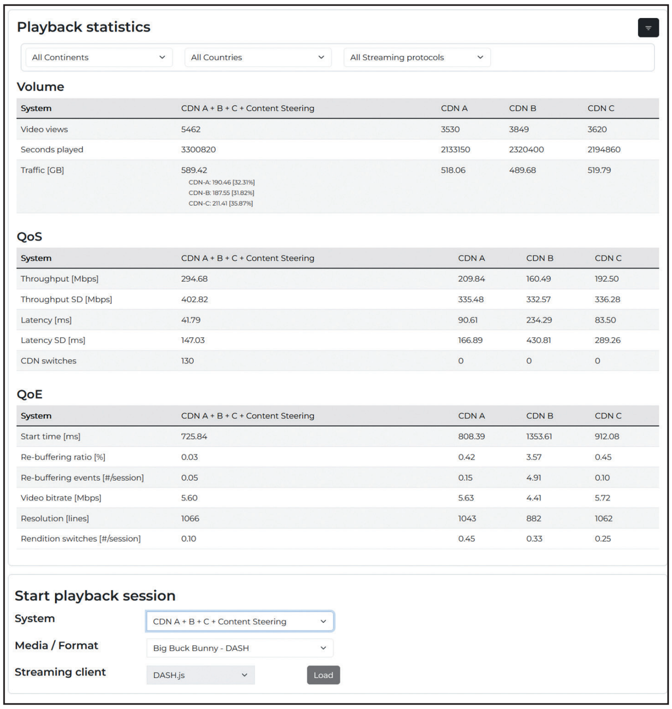


FIGURE 4. Information and controls on the main page of the content steering testbed.

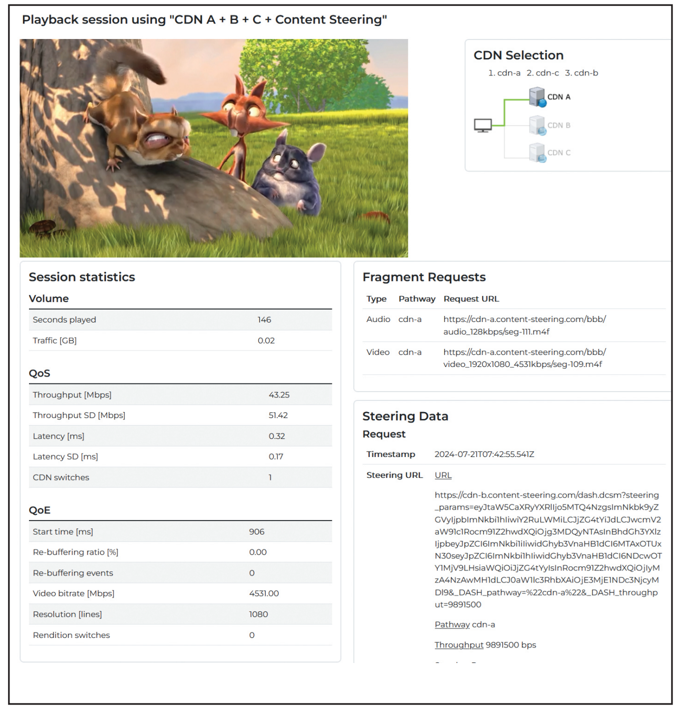


FIGURE 5. Playback session information in the content steering testbed.

Table 4. As a DASH player, the testbed employs the DASH.js player.[27] As an HLS player, the testbed employs HLS.js.[35]

Testbed Operation

The main testbed web page is available at https://testbed.content-steering.com. It includes the playback statistics dashboard and a tool for launching new streaming sessions. We present a screenshot of this page in **Fig. 4.**

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 using 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 playback statistics, as shown in **Fig. 5.**

The testbed allows users to launch many sessions on different devices and from all possible regions in the world. As players play the videos, they periodically send observed metrics to the QoE metrics processing engine implemented by the testbed. It stores all received and processed metrics in the database. The playback statistics reported on the main testbed page (cf. **Fig. 4)** represent the summary statistics based on data collected thus far.

Test Results

As we can observe from **Fig. 4,** the testbed currently reports the execution of over 5,000 sessions using content steering technology and over 16,000 sessions overall. The overall playback time of these sessions is about 3000 hr, and the overall volume of media data delivered is over 2000 Gbytes.

The statistics panel reports that overall, for a system with content steering, the current effective distribution of traffic between the three CDNs is CDN-A: 32.31%, CDN-B: 31.82%, and CDN-C: 35.87%. It is close to the even split, as set by the target per-CDN distribution. However, the traffic distributions in each region can be very different.

For example, if we look at statistics for India, shown in **Fig. 6,** we notice that measured throughputs of CDNs in this region are not so great. They amount to 36.6, 8.27, and 47.27 Mbits/s for CDN-A, B, and C, respectively. The measured latencies are suboptimal, particularly for CDN-B. Based on these metrics, the CDN-C appears to be the best CDN choice in this region, while CDN-B is the worst.

Next, let us look at the effective CDN load allocations achieved by the steering system in this region. They are reported as 75.9, 17.4, and 6.66% for CDNs A, B, and C, respectively. In other words, this system allocates most traffic to the better-performing CDNs in this region.

We further note that the steering system has executed seven mid-stream CDN switches for 131 playback sessions. Such switches happen when edge steering servers determine that continuing playback with the current CDN is impossible.

The effects of such steering decisions can be immediately appreciated by looking at the QoE statistics in **Fig. 6**. Here, we see that the system with three CDNs and steering delivers significantly better performance than the

1545-0279/25©2025SMPTE

ones reported for single CDNs. We note that even in comparison with the best CDN in this region (CDN-C), the system with steering achieves notable improvement. It reports the effective re-buffering rate of 0.14% vs 1.81% achieved by a system using only CDN-C. The improvements relative to the other CDNs are even more impressive.

Figure 7 presents an example of a region (France) where this system allocates traffic differently. In this region, the performance of all 3 CDNs is good. They all deliver at least 140 Mbits/s in throughput, and their latencies are less than 27ms. While CDN-B in this region is still not the best in throughput, it is more than good enough, and the system routes almost all traffic to it. This routing decision explains how this steering system balances the traffic on a multi-regional scale. It moves traffic away from underperforming CDNs in some regions and loads them more in regions where their performance is adequate.

As we look at the overall statistics for all regions, as shown in **Fig. 4,** we notice that the system with content steering achieves almost even distribution of traffic between three CDNs, and it also notably improves QoE. 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 a multi-CDN system with content-steering technology significantly outperforms all other systems in terms of reported QoE metrics.

A more detailed comparison of these systems, also utilizing ITU-T P.1203 parametric quality assessment metric [51], as well as subjective tests, is currently progressing in the SVTA players and playback SG.

Conclusion

In this paper, we have reviewed the content steering standard. We have explained its operating principles, features, and benefits. We have also surveyed the available implementations of streaming clients, servers, and various additional tools supporting this standard and the ongoing efforts of DASH-IF and SVTA organizations to support the rollout of this technology in the industry. We have also presented the results of a recent SVTA performance study, validating the benefits of this technology. With such encouraging results, a vast selection of clients, servers, validation tools already available, and continued support by the DASH-IF and SVTA organizations, this standard is well under way toward industry adoption.

Playback statistics

All Continents | India | All Streaming protocols

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. Summary playback statistics as observed in India.

Playback statistics

All Continents | France | All Streaming protocols

Volume

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Video views	272	148	177	138
Seconds played	164962	89936	107333	82577
Traffic [GB]	27.23 CDN-A: 0.48 [1.77%] CDN-B: 26.29 [96.52%] CDN-C: 0.47 [1.71%]	16.33	23.18	13.02

QoS

System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C
Throughput [Mbps]	312.89	186.37	165.64	142.39
Throughput SD [Mbps]	454.87	264.07	293.98	203.92
Latency [ms]	16.70	22.97	19.31	26.29
Latency SD [ms]	62.32	52.98	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	376.24	642.87	391.55
Re-buffering ratio [%]	0.00	0.00	0.00	0.00
Re-buffering events [#/session]	0.00	0.00	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 7. Summary playback statistics as observed in France.

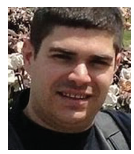
References

1. D. Wu, Y.T. Hou, W. Zhu, Y-Q. Zhang, and JM Peha, Streaming Video over the Internet: Approaches and Directions, *IEEE Trans. CSVT*, 11(3): pp. 282-300, 2001.
2. B. Girod, M. Kalman, Y.J. Liang, and R. Zhang, "Advances in Channel-adaptive Video Streaming,"*Wireless Comm. and Mobile Comp.,"* 2(6): 573-584, 2002.
3. G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman, and Y. A. Reznik, "Video Coding for Streaming Media Delivery on the Internet," *IEEE Trans. CSVT*, 11 (3): 269-281, 2001.
4. B. T. Bentaleb, A. C. Begen, C. Timmerer, R. Zimmermann, A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP, *IEEE Communications Surveys & Tutorials*, 21(1): 562-585, 2019.
5. Y. A. Reznik, K. O. Lillevold, A. Jagannath, and X. Li. 2021, "Towards Understanding of the Behavior of Web Streaming, Proc. Picture Coding Symposium (PCS)," pp. 1-5, 2021.
6. Y. Reznik, X. Li, K.O. Lillevold, R. Peck, T. Shutt, and P. Howard, Optimizing Mass-Scale Multi-Screen Video Delivery, SMPTE Motion Imaging Journal, 129 (3): 26-38, Apr. 2020.
7. R. Pantos, and W. May, HTTP live streaming, IETF, RFC 8216, https://tools.ietf.org/html/rfc8216, 2017.
8. ISO/IEC 23009-1:2022, Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media presentation description and segment formats, ISO/IEC, Oct. 2022.
9. Mind Commerce, CDN Market by Technology, Platform, Application, Service Type, Customer Type, and Industry Verticals 2021-2027, 2021.
10. EBU TR 068, CDN Architectures Demystified, EBU, Geneva, June 2022. [Online]. Available: https://tech.ebu.ch/publications/tr068
11. D. Hassoun, How to Jump-Start Your Multi-CDN Strategy and Deliver Every Time, October 2019. [Online]. Available: https://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=134765

12. Muvi.com, Multi-CDN switching methods, October 2019. Online: https://www.muvi.com/blogs/multi-cdn-switching-in-streaming-businesses.html
13. SVTA, SVTA investigation of approaches to CDN delivery, January 2023. [Online] Available: https://www.svta.org/2023/01/03/investigating-approaches-to-multi-cdn-delivery
14. Y. Reznik, G. Cabrera, On Multi-CDN Delivery Costs Optimization Problem, *Proc. IEEE International Symposium on Multimedia (ISM)*, Tokyo, Japan, Dec. 11-13, 2024.
15. R. Pantos, RFC 8216bis, HTTP Live Streaming 2nd Edition, Section 7: Content steering, v10, IETF, Nov. 2021. [Online] Available: https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis#section-7
16. ISO/IEC DIS 23009-1: Information technology — Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats, ISO/IEC, July 2024. [Online] Available: https://www.iso.org/standard/89027.html
17. ETSI TS 103 998, DASH-IF: Content Steering for DASH, v1.1.1, ETSI, Jan. 2024, [Online] Available: https://www.etsi.org/deliver/etsi_ts/103900_103999/103998/01.01.01_60/ts_103998v010101p.pdf
18. Y. Reznik, G. Cabrera, and A. Waldron, "Simplifying Multi-CDN Delivery with HLS / DASH Content Steering," presented at the SMPTE Media Technology Summit, Hollywood, CA, Oct. 16-19, 2023.
19. Y. Reznik, G. Cabrera, D. Silhavy, S. Pham, A. Giladi, A. Balk, A. C. Begen, and W. Law, "Content Steering: a Standard for Multi-CDN Streaming," *Proc. ACM Mile-High Video Conference (MHV '24)*, Denver, CO, Feb. 11-14, 2024
20. D. Silhavy, S. Pham, A. Giladi, A. Balk, A. Begen, and W. Law, Real-Time Streaming Reliability and Performance Optimization Using Content Steering, presented at the SMPTE 2023 Media Technology Summit, Hollywood, CA, Oct. 16-19, 2023.
21. E. Gama, R. Rodrigues-Filho, E. Madeira, R. Immich, and L. F. Bittencourt, Enabling Adaptive Video Streaming via Content Steering on the Edge-Cloud Continuum, *Proc. IEEE International Conference on Fog and Edge Computing (ICFEC 2024),* Philadelphia, PA, May 06-09, 2024.
22. Apple, HLS Content Steering Specification (v1.2b1), Apple, April 2021. [Online] Available: https://developer.apple.com/streaming/HLSContentSteeringSpecification.pdf
23. DASH-IF, 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
24. Consumer Technology Association (CTA), CTA-5004: Common Media Client Data (CMCD), CTA, Sep. 2020, https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5004-final.pdf
25. A. Bentaleb, M. Lim, M. N. Akcay, A. C. Begen, and R. Zimmermann, Common media client data (CMCD): initial findings, *Proc. ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '21),* pp. 25-33, 2021.
26. Streaming Video Technology Alliance (SVTA), Open Caching Project. [Online] Availabilty: https://opencaching.svta.org/
27. DASH-IF, Reference client, online: https://dashif.org/tools/dashjs/
28. DASH-IF, Test and Reference tools, online: https://dashif.org/tools/introduction/
29. DASH-IF, Content Steering demo & test stream, online: https://reference.dashif.org/dash.js/nightly/samples/advanced/content-steering.htm
30. Streaming Video Technology Allicance (SVTA), Architectures for Multi-CDN Switching, December 2021, https://www.svta.org/project/architectures-for-multi-cdn-switching
31. Streaming Video Technology Allicance (SVTA), Content Steering Prototype, online: https://github.com/streaming-video-technology-alliance/content-steering-prototype (available to SVTA members)
32. Streaming Video Technology Alliance (SVTA), Content Steering at Edge, online: https://github.com/streaming-video-technology-alliance/content_steering_at_edge (available to SVTA members)
33. Apple, AVFoundation. [Online] Available: https://developer.apple.com/av-foundation/
34. Apple, HLS developer tools. [Online] Available: https://developer.apple.com/streaming/
35. Hls.js player. [Online]. Available: https://github.com/video-dev/hls.js/
36. Video.js player. [Online] Available: https://videojs.com/
37. Shaka player. [Online] Available: https://github.com/shaka-project/shaka-player
38. Brightcove player. [Online]. Available: online: https://www.brightcove.com/en/products/player/
39. Bitmovin video player. [Online]. Available: https://bitmovin.com/video-player/
40. Radiant media player. [Online] Available: https://www.radiantmediaplayer.com/
41. ExoPlayer. [Online] Available: https://github.com/google/ExoPlayer
42. DASH-IF, Content steering POC using shaka packager and exoplayer. [Online] Available: https://github.com/Dash-Industry-Forum/Content-Steering/issues/17
43. ExoPlayer ticket: Add support for HLS/DASH content steering. [Online] Available: https://github.com/google/ExoPlayer/issues/11203
44. Shaka Packager. [Online] Available: https://github.com/shaka-project/shaka-packager
45. Shaka Packager ticket: Add content steering support. [Online]. Available: https://github.com/shaka-project/shaka-packager/issues/1403
46. EINBLIQ.IO, Content Steering Server. [Online] Available: https://einbliq.io/einbliq-io-content-steering-server/
47. Brightcove Content Steering @ Edge demonstration. [Online] Available: https://www.content-steering.com/
48. SVTA Content Steering testbed. [Online] Available: https://testbed.content-steering.com/
49. Y. Reznik, G. Cabrera, R. Zekarias, B. Zhang, B. Panigrahi, N. Barman, S. Hicks, T. Krofssik, A.Sinclair, and A. Waldron, Implementing HLS/DASH Content Steering at Scale, Proc. International Broadcast Convention (IBC 2023), Sep. 15-18, 2023.
50. Blender Foundation, Big Buck Bunny video sequence. [Online]. Available: https://peach.blender.org/
51. International Telecommunication Union-Telecommunication (ITU-T) P.1203: Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport, ITU-T, Oct. 29, 2017. [Online]. Available: https://www.itu.int/rec/T-REC-P.1203-201710-I/en
52. Y. Reznik, G. Cabrera, D. Silhavy, S. Pham, A. Giladi, A. Balk, A. C. Begen, and W. Law, "Content Steering: a Standard for Multi-CDN Streaming," *Proc. International Broadcast Convention (IBC 2023),* Sep. 15, 2024.

About the Authors

Yuriy Reznik is a Technology Fellow and Vice President of Research at Brightcove, Inc. He has worked on streaming since 1997, contributing to several Emmy-winning products and standards.



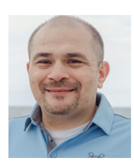
Guillem Cabrera is a principal software engineer at Brightcove U.K., Ltd., leading the development of advanced multi-CDN delivery technologies. Before Brightcove, Cabrera was a software engineer with VideoAsk / Typeform (2021- 2023) and Rakuten TV (2015-2021).

Daniel Silhavy is a scientist and project manager at the Future Applications and Media (FAME) Business Unit of the Fraunhofer Institute for Open Communication Systems (FOKUS), Berlin, Germany. He is also the lead developer of the dash.js project and the development coordinator for the 5G-Media Action Group (5G-MAG) Reference Tools.

Stefan Pham is a project manager at the FAME Business Unit of FOKUS, Berlin, Germany. He specializes in R&D topics dealing with internet-delivered media, TV, and cross-platform web apps.

Alex Giladi is a Comcast Fellow and an Emmy-winning technologist. He has been working on issues related to video content encoding and distribution since 2000. His current areas of interest are video encoding and adaptive streaming.

Alex Balk is a senior software engineer at Comcast, working on advanced research projects in cloud computing, video encoding, and security. Previously, he was a software engineer with Brightcove, InterDigital, Vibuiquity, Rockwell Collins, and SAIC.



Ali C. Begen is currently a computer science professor at Ozyegin University and a technical consultant in Comcast's Advanced Technology and Standards Group. Previously, he was a research and development engineer at Cisco.

Will Law is a chief architect within the Cloud Technology Group at Akamai and a leading media delivery technologist involved with streaming media on the internet for the past 20 years.

Presented at IBC 2024. Amsterdam, The Netherlands, 13-16 Sept. 2024. This paper is published here with the kind permission of the IBC. Copyright © IBC.

DOI: 10.5594/JMI.2025/NNRR2188
Date of publication: 24 January 2025