# SEGMENTS: 2024

A CARLES AND A DATA OF A D

DELIVERY

# **ARCHITECTURES FOR MULTI-CDN SWITCHING**

Yuriy Reznik VP, Research Brightcove, Inc.

**#SEGMENTS2024** 

FEBRUARY 14, 2024 | DENVER MARRIOTT TECH CENTER



## OUTLINE

- Overview of activity in SVTA P&P SG
- Technologies under consideration / study 2.
  - Client-side switching
  - Server-side content steering
  - Edge-supported steering framework
- Demo & Preliminary results 3.
- Conclusions 4.

SEGMEN	<b>IS:</b> 2024
C      Architectures for Multi-CDN Swit: x +     C      https://www.svta.org/project/architectures-for-multi-cdn-switching/     A	3   D ½ @ % Ø
Join Work V Sites V Members V Education V Events V About V	Y ÎL (B)
······	
Architectures for Multi-CDN S	Switching
>>>>> in	Players and Playback
Project Status: Working Start: December 1, 2022 Estimated Com	npletion: <sub>January 31, 2024</sub>
Home > Architectures for Multi-CDN Switching	
Problem Statement	Project Type
issues with CDNs can impact QoE at the player if a specific CDN is performing poorly but still being used to fulfill the requests. The delivery system needs to include business logic and heuristics to be able to switch between CDNs, for whatever reason, and have the requests appropriately modified to reflect the new CDN UR.	Code
Project Description	Project Leads

Ali C. Begen Multi-CDN delivery has become commonplace in scalable streaming video architectures, but switching to a different CDN in real time to improve QoE is not easy. Many server-side and client-side solutions are available to BRIGHTCOVE Yuriy Reznik streaming operators, but there is little consensus on the best way to accomplish it. This project studies several existing and proposed solutions to this problem to understand their merits and scenarios when each can be recommended. The Load Balancer is a client-side solution proposed in this project. It is a TypeScript library that serves as a client-side multi-CDN switcher for video streaming. For each video segment requested by the player, Advisors the Load Balancer determines the best CDN to use and outputs a modified HTTP request identifying that determined CDN. The Load Balancer determines the best CDN by maximizing business rules and OoS scores with initial values provided by an external interface. Once playback begins, an internal algorithm in the Load Balancer updates the QoS scores with each content delivery network request. The Load Balancer repository also comes with a demo page serventer Daniel Silhavy The demo page shows lightweight JavaScript integration of the Load Balancer with the open-source Shaka player. Content steering is a combined client-side and server-side architecture relying on Content steering functions of the HLS and DASH standards. There have been two proposed implementations of such solutions to the project. One



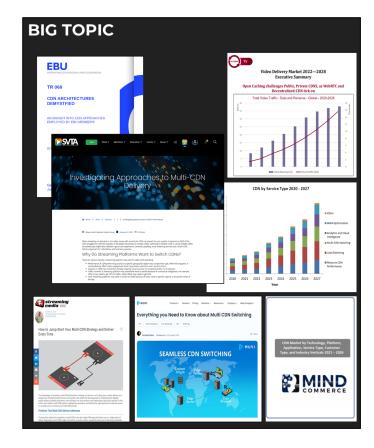
# MULTI-CDN DELIVERY

#### **SEGMENTS:**2024

# Many architectures have been proposed in the past Architecture Pros Cons This is the simplest of all solutions since the Switch delay is more time-consuming, ranging from

DNS-based	This is the simplest of all solutions since the source video URL always remains constant.	Switch delay is more time-consuming, ranging from 300 seconds to even five minutes in case of CDN failures. This can immensely hamper the user QoE.
On-the-fly manifest rewrite	Better user experience due to midstream switching eliminating the need for hard refresh during video playback. No matter the volume of session resets, this method reduces the chances of a cascade effect that may hamper the video workflow.	Rewriting the manifest can sometimes bring about errors. Midstream switching is not completely seamless, and 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 since changes happen in the server itself that is 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 necessarily consider the unique conditions of the actual clients.
Client-side	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 that requires detailed planning.

https://www.svta.org/2023/01/03/investigating-approaches-to-multi-cdn-delivery/



#### But none of them is perfect...

- The objective of SVTA activity is to study and produce report guiding the industry on the available choices



# **ARCHITECTURES UNDER STUDY**



SVTA member contributions:

#### Client-side solution:

CDN Load Balancer (Lumen)

	pgy-allia / cdn-load-balanc A Q Type [7] to s	earch	> + • O n Ø
	requests ③ Actions	① Security \u2222 Insi	
cdn-load-balancer		⊙ Secondy L India	
알 main ▼ 알 1 Branch ⓒ 0	Tags Q Go to file t	+ Code +	About
Sergears Merge pull request	#6 from streaming-video-technol 🚥 Sc78938 · last	year 🕚 11 Commits	Midstream multi-CDN switcher for intelligent video load balancing
github	Add codeowners	2 years ago	🛱 Readme
bemo	Increase shaka retry maxAttempts	last year	बी्व MIT license ⊸∽ Activity
img	Initial release	2 years ago	<ul> <li>Custom properties</li> </ul>
lib	Initial release	2 years ago	☆ 4 stars
test	Initial release	2 years ago	<ul> <li>0 watching</li> <li>0 forks</li> </ul>
.editorconfig	Initial release	2 years ago	
.eslintrc	Initial release	2 years ago	Releases
.gitignore	Initial release	2 years ago	Create a new release
.prettierrc	Initial release	2 years ago	Packages
CHANGELOG.md	Initial release	2 years ago	No packages published
	Create LICENSE	2 years ago	Publish your first package
	Update README.md	2 years ago	Contributors 3
README.md			
README.md jest.config.js	Initial release	2 years ago	Sergears Sergey Arsenyev

https://github.com/streaming-video-technologyalliance/cdn-load-balancer

#### Server-side solution:

Content steering prototype (Comcast)

streaming-video-technology-all:	× +			- 0
C 🗅 https://github.com/strea	ming-video-technology-alliance	/content-steering-prototyp	e ⊞ A%	☆ ©   Φ ☆ ⊕ ⊗ …
Streaming-video-techno /	content-steering-prototy 👌	Q Type 🛛 to search		)>   + • O n 🗠 🤅
Code 💿 Issues 🏦 Pull requests		🕮 Wiki 💿 Security	y 🗠 Insights	
content-steering-prototy;	Private		Watch 0	▼ <sup>1</sup> Fork 0 ▼ ☆ Star 1 ▼
₽ master → 우 1 Branch ⓒ 0 Tag	Q Go to file	t +	⇔ Code 👻	About
ab2022 update priority name		47b9288 · 10 months ago	6 Commits	No description, website, or topics provided.
endpoint	add support for PATHWA	-PRIORITY and L 10	) months ago	-∿ Activity
nginx-consteer-module	differentiate between BAS	EURL and LOCATI 10	) months ago	E Custom properties
LICENSE-2.0.txt	add license		last year	☆ 1 star ⊙ 0 watching
🗋 Readme.md	update priority name	10	) months ago	😵 0 forks
README			Ø :≡	Releases
				No releases published Create a new release
Comcast server side (	Content Steering p	rototype		
This repository contains a workin	n server side implementation	of CDN content steeri	ng The	Packages
functionality was validated with o described below.				No packages published Publish your first package
Part 1. NGINX module				Languages
The NGINX module adds the xml manifests.	elements needed enable co	ntent steering to DASH		• C++ 99.4% • Go 0.6%
Part 2. Steering Endpoint				
	ering endpoint API. It respo			

https://github.com/streaming-video-technologyalliance/content-steering-prototype

#### Edge-side solution:

Content steering @ edge (Brightcove)

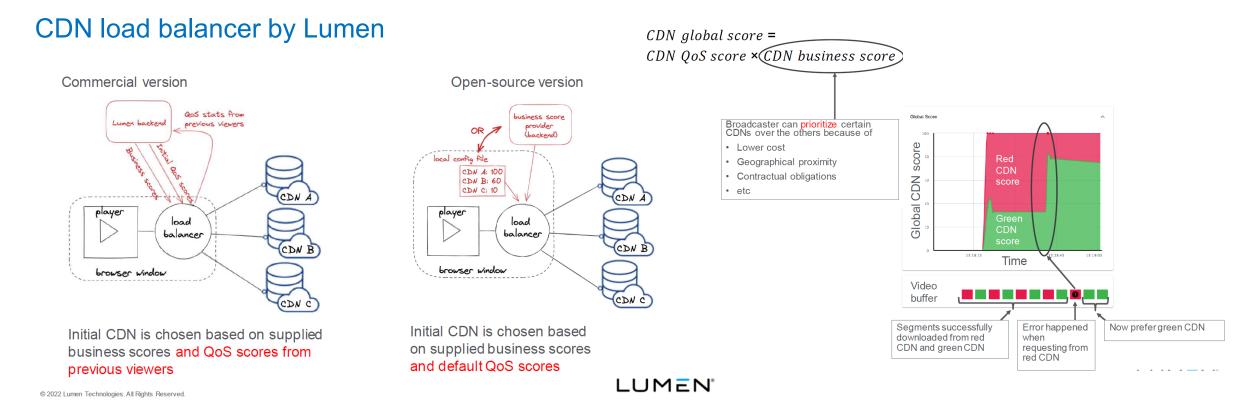
Streaming-video-technol	o / content_steering_at_ed 🛆			
	, content_steering_account B	Q Type [] to search		>_ + • ⊙ tì 🖻
Code 💿 Issues 📫 Pull requ	uests 🕑 Actions 😲 Security	🗠 Insights		
content_steering_at_ed	ge Private		⊙ Watch 0	▼ V Fork 0 ▼ ☆ Star 1 ▼
P main ▼ P 1 Branch 🛇 0	Tags Q. Go to file	t +	↔ Code →	About
guillemcabrera Using multipl	e reported pathways & throu 🚥	9057ec - 6 months ago	🕚 7 Commits	This repository contains the original Content Steering @ Edge proposal from
manifest-updater	Add manifest-updater com	oonent	last year	Brightcove
steering-server	Using multiple reported pa	hways & throug	6 months ago	cdn-switching
LICENSE.md	Add License		last year	🛱 Readme
README.md	Add repo README		last year	View license
README A License			/ ≔	<ul> <li>小 Activity</li> <li>I Custom properties</li> </ul>
				☆ 1 star
Content Steer	ing @ Edge			<ul> <li>⊙ 0 watching</li> <li>♀ 0 forks</li> </ul>
	nplementation of CDN switching	echnoloav utilizina	Content	Languages
Steering mechanisms of HLS		5, 5		
effectively a function that ma	olementation is a stateless operat w be executed once per server re- cost-effective deployment using e forms.	uest. Such impleme	intation	Go 53.9% JavaScript 44.7%     Dockerfile 1.4%
	lementation is a much lower TTL i aming players. This enables steeri ng behavior, and perform context	g servers to react fa		

https://github.com/streaming-video-technologyalliance/content\_steering\_at\_edge



### **CLIENT-SIDE SWITCHING**

#### **SEGMENTS:**2024



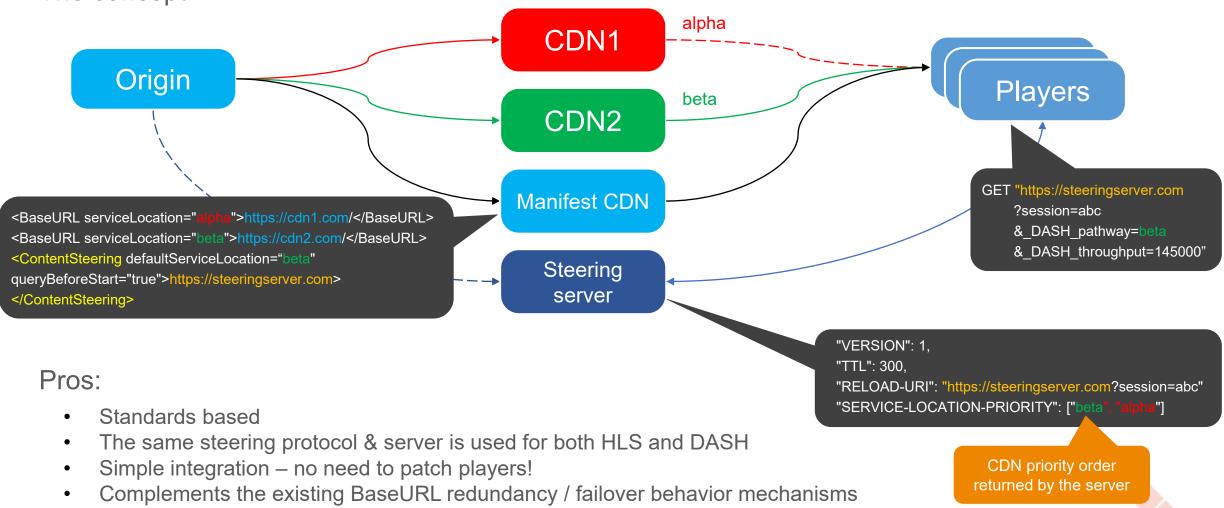
#### SVTA open source version works with:

- Shaka player (HLS, DASH), and DASH.js (DASH)
- Requires back-end for providing initial QoS and business scores.

# HLS / DASH CONTENT STEERING

#### **SEGMENTS:**2024

The concept



# **CONTENT STEERING: CLIENTS**



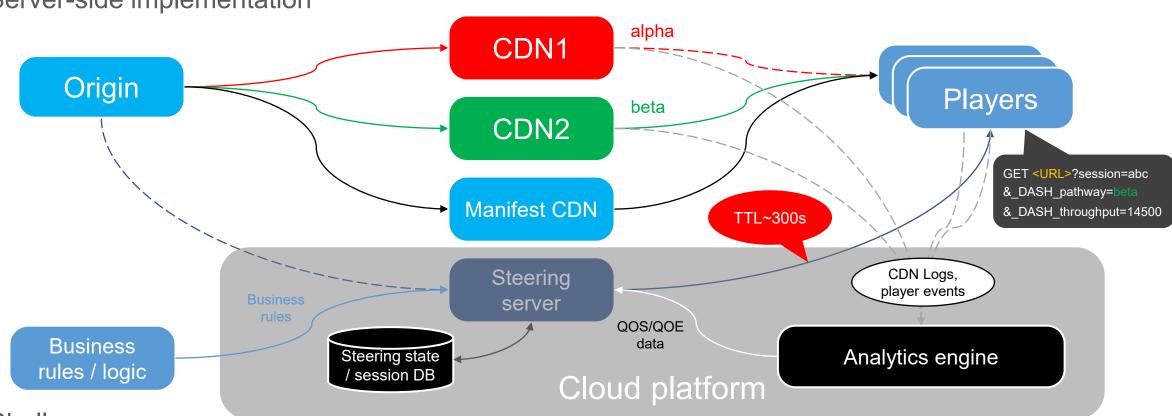
Content-steering-capable clients:

Media Player	DASH Content Steering	HLS Content Steering
<b>dash</b> .js	since version 4.5.0	
<b>VIDEO</b> .JS	since version 8.8.0	since version 8.8.0
Shaka <sub>Player</sub>	since version 4.6.0	since version 4.6.0
G ExoPlayer	planned for 2024	planned for 2024
<b>hls</b> .js		since version 1.4.0
🗳 AVPlayer		since iOS version 15.0



# **CONTENT STEERING SERVER**

#### **SEGMENTS:**2024



Server-side implementation

#### Challenges

- TTL time: 300s default is too long! Suitable for basic CDN load balancing. Not suitable for QOE optimizations!
- Scalability: the steering server should be at least as scalable as manifest CDN!
- Costs: reducing TTL will increase number of requests and traffic to the steering server!

# CONTENT STEERING @ EDGE

#### **SEGMENTS:**2024

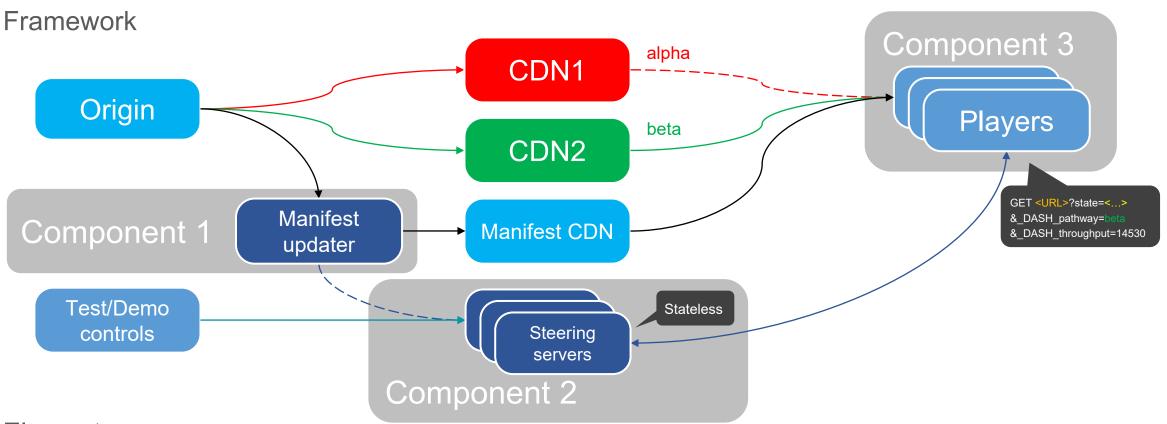
Edge-based architecture alpha CDN1 Origin Players beta CDN2 GET <URL>?state=<...> & DASH pathway= Manifest & DASH throughput=14530 **Manifest CDN** updater TTL~10..30s CDN Logs, Steering Business Steering player events master rules servers @ edge Business Stateless Analytics engine rules / logic Steering DB Adv. CDN or edge platform platform Clo QOS/QOE data

#### Benefits

- Scales well with CDNs or edge platforms.
- TTL can be smaller; comparable to player buffer delay; Can be used to optimize QOE!

# **OPEN SOURCE PROJECT IN SVTA**

#### **SEGMENTS:**2024

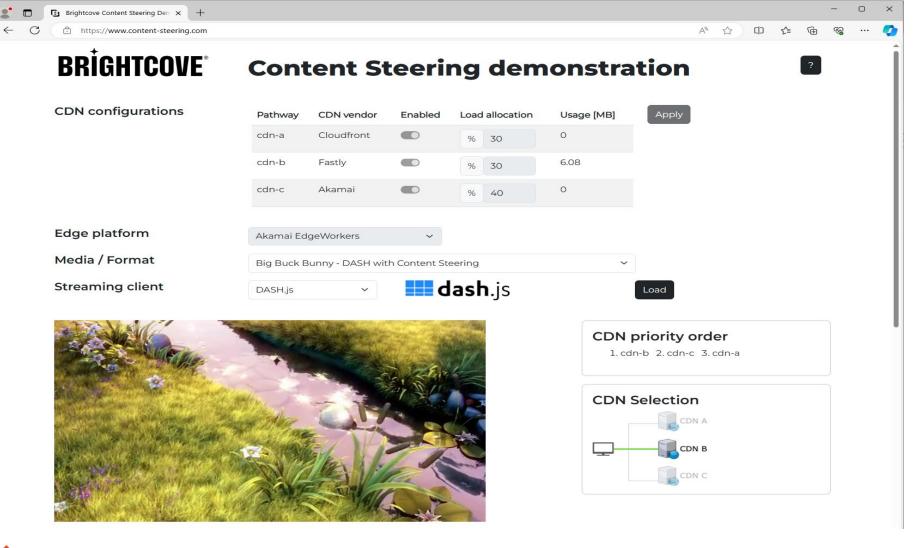


Elements:

- Component 1: manifest updater inserting content steering information in the manifests (Golang)
- Component 2: steering server implementation with several deployment variants (Node.js, Lambda @ Edge, etc.)
- Component 3: DASH and HLS players: DASH.js, HLS.js, video.js, shaka existing open source projects

## DEMO: www.content-steering.com







# **TESTING FRAMEWORK**

#### Control panel:

Playback statistic	3				
Volume					
System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C	
Video views	615	581	581	587	
Seconds played	368432	351974	350868	345320	
Traffic [GB]	54.41	53.46	53.42	53.72	
QoS					
System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C	
Throughput [Mbps]	199.91	217.72	156.79	209.88	
Throughput Std Dev [Mbps]	242.06	247.96	238.23	248.50	
Latency [ms]	20.08	12.98	39.90	21.84	
Latency Std Dev [ms]	95.02	141.01	1483.71	547.87	
CDN switches	38	0	0	0	
QoE					
System	CDN A + B + C + Content Steering	CDN A	CDN B	CDN C	
Start time [ms]	797.09	631.74	669.14	909.09	
Re-buffering ratio	0.57	0.59	1.33	0.61	
Re-buffering events	169	172	172	173	
Video bitrate [Mbps]	3.46	3.44	3.42	3.41	
Resolution [lines]	1558	1551	1546	1541	
Rendition switches	664	617	603	662	

#### **Playback sessions** Playback session using "CDN A"



**CDN Selection** 1. cdn-a

CDN C

https://cdn-a.content-steering.com/bbb/ audio\_128kbps/seg-69.m4f https://cdn-a.content-steering.com/bbb/ video\_1920x1080\_4531kbps/seg-67.m4f

https://cdn-a.content-steering.com/dash.do m?steering\_params=evJtaW5CaXRvYXRIIio5 MTQ4NzgslmNkbk9yZGVyljpblmNkbilhll0sl nByZXZpb3VzVGhyb3VnaHB1dCl6Nzc4NTAv MCwicGF0aHdheXMiOlt7ImlkljoiY2RuLWEiL CJ0aHJvdWdocHV0ljo3OTUxOTQxfV0sInRpl WVzdGEtcCI6MTcwNzkzMTEzNiM0OX0%3E &\_DASH\_pathway=%22cdn-a%22&\_DASH\_ti

2024-02-14T17:19:07.440Z

roughput=4225250 Pathway cdn-a

Throughput 4225250 bps

Session statistics		Fragment	Requests
olume		Type Pathw	ay Request U
Seconds played	68.14	Audio cdn-a	https://cdn audio_128k
Fraffic [GB]	0.01	Video cdn-a	https://cdn
20S		video carra	video_1920
Throughput [Mbps]	10.59		
Throughput Std Dev [Mbps]	1.22	Steering D	ata
Latency [ms]	0.22	Request	
Latency Std Dev [ms]	0.08	Timestamp	2024-02-14T
CDN switches	0	Steering URL	URL
QoE			https://cdn-a m?steering_ MTO4NzasIr
Start time [ms]	744		nByZXZpb3\
Re-buffering ratio	0.00		MCwicGF0a CJ0aHJvdW
Re-buffering events	0		WVzdGFtcC &_DASH_pa
Video bitrate [Mbps]	4531.00		roughput=4
Resolution [lines]	1920		Pathway cdr
Rendition switches	0		Throughput
			Steering Par



#### **SEGMENTS:**2024

#### Statistics collection

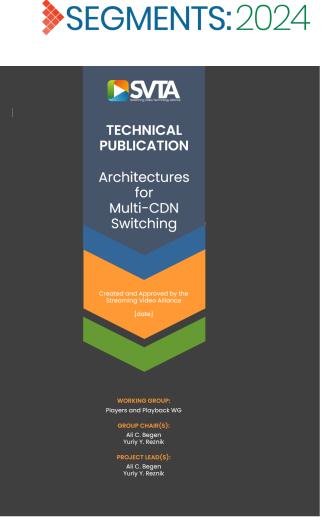


100000x

Real-world tests Multiple sites / labs No artificial throttling The same conditions to different technologies

# NEXT STEPS

- 1. Complete study / testing
- 2. Publish SVTA report on CDN switching
  - Problem explanation
  - Comparison of architectures
  - Summary of findings based on out tests
- 3. Synergies with other projects:
  - SVTA Open caching
  - SVTA QOE SG
- 4. Applications in hybrid delivery ecosystems
  - Wireless: 5G-MAG, 5G broadcast, etc.
  - Satellite assisted delivery: 5G-EMERGE, etc.





# SEGMENTS: 2024



#SEGMENTS2024