MULTI-CDN STREAMING: ARCHITECTURES AND OPTIMIZATION PROBLEMS

Yuriy Reznik Brightcove, Inc. Boston, MA, USA



ACM MMsys

March 31- April 4, 2025

Stellenbosch Inst for Adv Study

Stellenbosch, South Africa

OUTLINE

- ► Introduction
- Architectures for CDN switching
 - > DNS-based methods
 - > Dynamic manifest updates
 - > Client-side methods
- Content steering technology in HLS & DASH
 - > Main principles
 - > Pathway cloning
 - > Uses of CMCD
 - > Edge-based deployment
 - > State of deployment & open-source tools
- Algorithms and optimization problems
 - > Distribution of traffic
 - > Managing CDN commitments and costs
 - > QOS/QOE-based optimizations
- Conclusions



MULTI-CDN MEDIA DELIVERY

Key benefits / utilities

- Multi-region delivery
- ▶ Better scale (load balancing)
- ► Improved reliability (failover)
- ► Improved QOE (QOS/QOE optimizations)
- ► Lower operating costs (?)

Challenges

- Not trivial to implement
 - > Switching delays
 - > Seamless mid-stream switching problem
 - > Analytics collection
- ► Not trivial to deploy and operate
 - > Manual vs automated models
 - > Optimization criteria
 - > Optimization algorithms
 - > Operating costs





ARCHITECTURES FOR MULTI-CDN SWITCHING

DNS-based switching
Manifest updates
Client-side techniques
Server-side techniques
HLS/DASH content steering



DNS-BASED CDN SWITCHING

Principle of operation

- Uses DNS servers to redirect traffic between different CDNs.
- Decisions about switch are produced based on analytics data collected either directly from active players, or from instrumented players (probes) running in different regions.

Examples

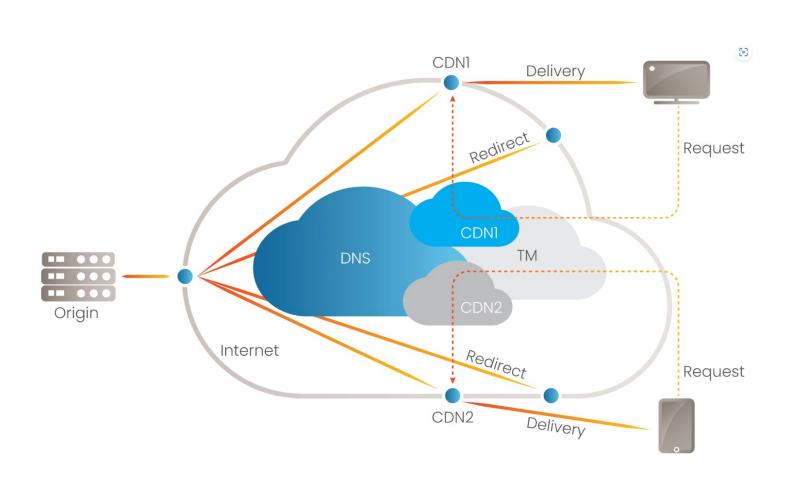
- ► NS1 (now IBM)
- Cedexis (now Citrix)
- Akamai GTM

Pros

- Relatively simple to implement.
- No changes in CMS, players, or servers as source video URL always remains constant.

Cons

- Switch delay is affected by DNS update propagation statistics.
- Practically it can range from several minutes to 1 hour (or even longer in some cases). This may affect QOE or availability of service if switching executed as a failover operation.
- ➤ Seamless in-stream switching is challenging. One must ensure exactly the same copies of content to be available on both CDNs, with identical sizes, timestamps, and availability timilines.





CLIENT-SIDE SWITCHING

Principle of operation

- Streaming client operates extra logic that decides when to switch between different CDNs
- Information about performance of alternative CDNs is obtained either by probing them occasionally or by pulling QOE statistics reported by other players in the same region (cross with P2P streaming).

Examples

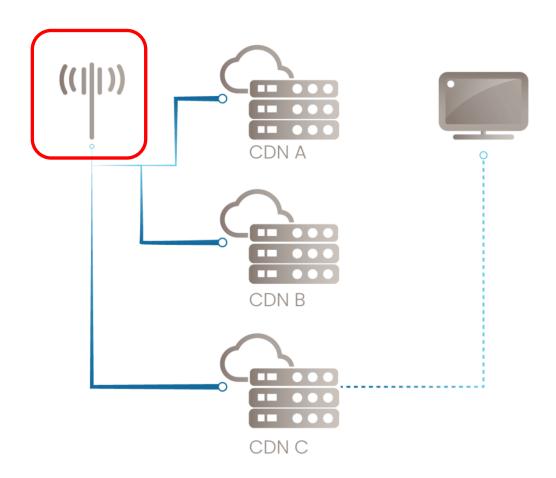
- StreamRoot / Lumen
- System73
- ► DVB-DASH profile

Pros

- QoS data is accurate as it is fetched based on individual clients' local and real-time performance metrics.
- Seamless midstream CDN switching may be possible.

Cons

- Requires updates of players.
- May not be feasible on some ("black box") platforms
- ► Maybe challenging to built in-house due to the code complexity of the algorithms that requires detailed planning, experimentation, and testing.

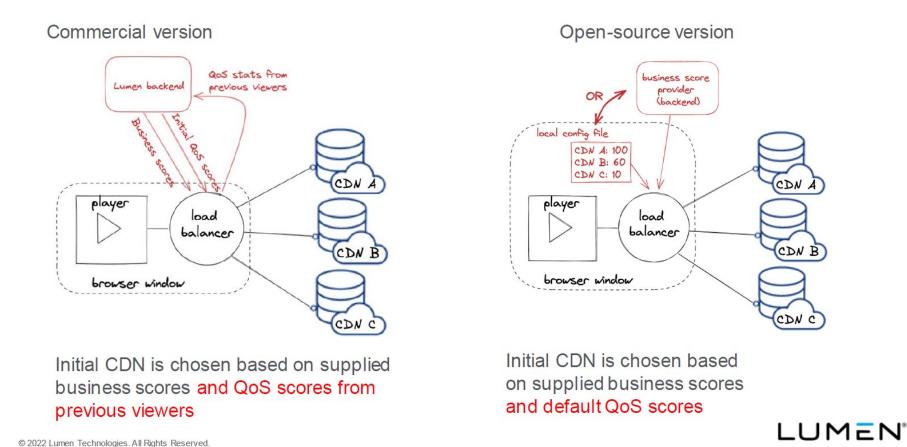


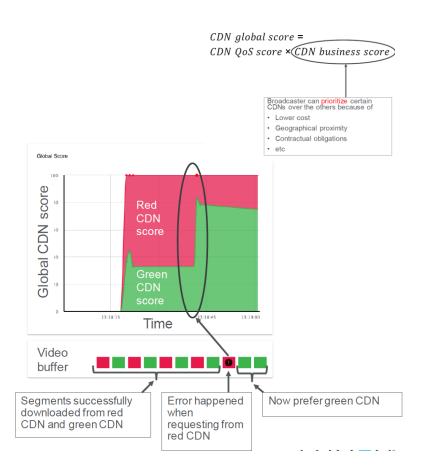


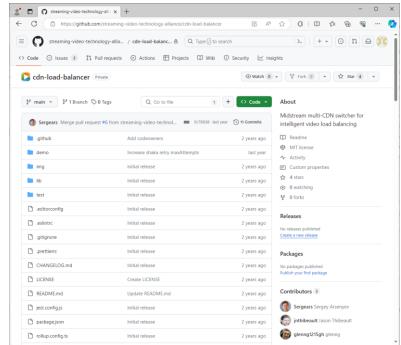
CLIENT-SIDE SWITCHING: EXAMPLE 1

CDN Load balancer by Lumen (StreamRoot)

► Limited ("SVTA Labs") implementation is available in open source







Limitations of SVTA labs version:

- ► Works only with Shaka player (HLS, DASH), and DASH.js (DASH)
- ► Requires back-end for providing initial QoS and business scores
- ► No longer supported by Lumen



CLIENT-SIDE SWITCHING: EXAMPLE 2

DVB-DASH profile

- ▶ Defined in DVB A168, Rev 6, 2022
- Enables clients to switch dynamically between different CDNs defined in the manifest
- ► Allows production workflows to define different CDN priorities and "weights"
- ► Defines algorithm (shaped random number generator) by which clients can select CDNs for given "weights"

Pros

Standards-based

Cons

- Limited to DASH
- Limited player support
- ► Requires manifest updates and logic in the delivery chain to define CND weights or priority orders

11.9.2 Prioritizing BaseURLs

BaseURLs containing absolute URLs can be assigned a priority using the @priority attribute. The value of this is a positive integer, with the lowest value indicating the highest priority. This allows additional BaseURLs to be specified to improve the reliability of a service, but on the basis that some of the BaseURLs are only to be used if higher priority ones cannot be used successfully.

The distance between values of @priority has no effect, so two BaseURLs with priorities of 1 and 2 would be the same as if they had values of 1 and 4.

It is possible for more than one BaseURL to have a given priority value, which allows load balancing to happen between the BaseURLs with that priority, as described in the following clauses.

As an example, if a content provider uses three CDNs, but they charge differently, with cheap-cdn being the cheapest and costly-cdn being the most expensive, that content provider may wish to use cheap-cdn most of the time, but make use of the others if there are problems. In that case they would create three BaseURLs, with the @priority attribute being used to signal the order in which they are used:

```
<BaseURL dvb:priority="1">http://cheap-cdn.example.com//BaseURL>
<BaseURL dvb:priority="2">http://moderate-cdn.example.com/</BaseURL>
<BaseURL dvb:priority="3">http://costly-cdn.example.com/</BaseURL>
```

This would mean that Players used cheap-cdn first, falling back to moderate-cdn if there is a problem with cheap-cdn. Costly-cdn would only be used if both cheap-cdn and moderate-cdn had problems.

11.9.3 Load balancing

Load balancing can often be carried out using either equipment in the network, or DNS load balancing. However such mechanisms may not be ideal for balancing between different hosting locations or CDNs, or where the load needs to be distributed in non-equal ratios.

A facility for load balancing is provided in clause 10.8.2, and a @weight attribute is added to the BaseURL element to allow the relative weighting of different BaseURLs to be signalled. The values of @weight are only significant relative to the values of other BaseURLs with the same @priority value. The BaseURL to use is chosen at random by the Player, with the weight of any given BaseURL being its @weight value divided by the sum of all @weight values.

As a simple example, consider a content provider uses two CDNs, cdn1 and cdn2 and wishes the load to be equally spread across them. This can be achieved by setting the @weight value on each to be the same, e.g.:

```
<BaseURL dvb:priority="1" dvb:weight="1">http://cdn1.example.com/</BaseURL>
<BaseURL dvb:priority="1" dvb:weight="1">http://cdn2.example.com/</BaseURL>
```

In a more complex example, consider a content provider using three CDNs, with one of them having a lower purchased capacity. The content provider needs to ensure that cdn3 only has 10 % of the load, with the remaining 90 % being shared equally by cdn1 and cdn2. They could do this by using the following entries:

```
<BaseURL dvb:priority="1" dvb:weight="9">http://cdn1.example.com/</BaseURL>
<BaseURL dvb:priority="1" dvb:weight="9">http://cdn2.example.com/</BaseURL>
<BaseURL dvb:priority="1" dvb:weight="2">http://cdn3.example.com/</BaseURL>
```



MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks

DVB Document A168 Rev.6

October 2022



SERVER-SIDE SWITCHING

Principle of operation

- In server-side switching, a specialized server analyzes data from various sources, such as the clients, CDN logs and synthetic agents.
- ► This data is then used to determine which CDNs to utilize for delivery, and individual manifests are modified for player sessions and pushed down to the appropriate clients.

Examples

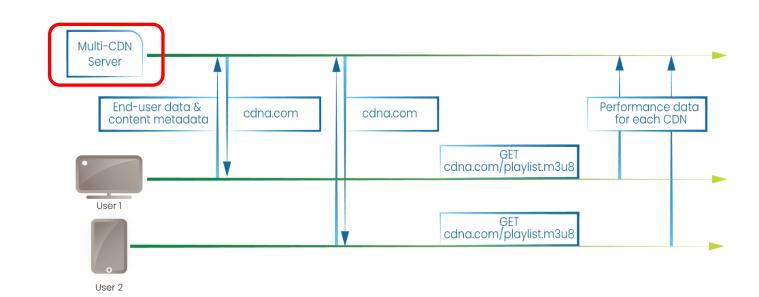
- ▶ DLVR
- ► HLS/DASH content steering standard

Pros

- Concentrates control in hands of an operator
- Relatively simple to implement

Cons

- Analytics data maybe delayed (by collection chain) and not specific for each client / session.
- Server-side processing also adds decision delays
- Implementing fine-grain accuracy switching comes with extra complexity and costs

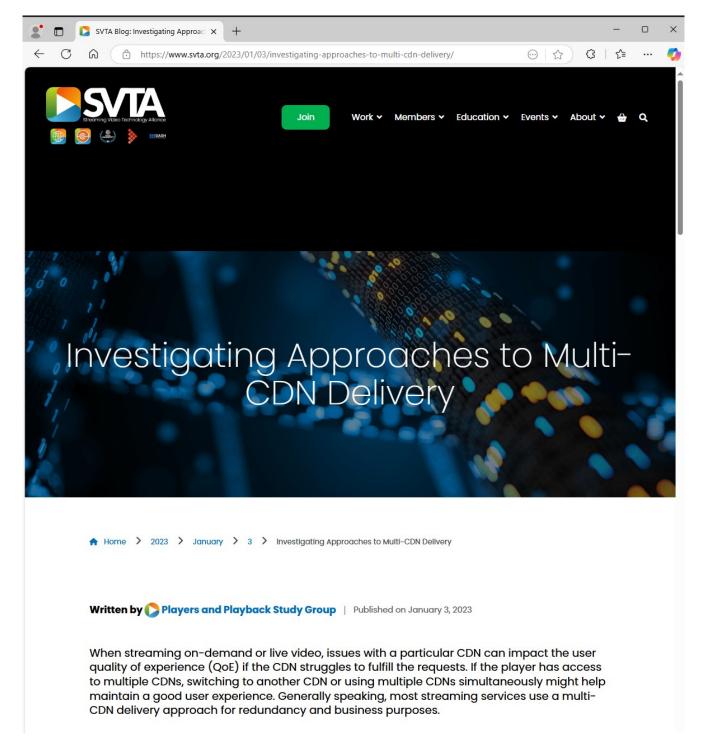




CDN SWITCHING ARCHITECTURES

A bigger picture

Architecture	Pros	Cons
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/

HLS/DASH CONTENT STEERING STANDARD

Principle of operation

Benefits

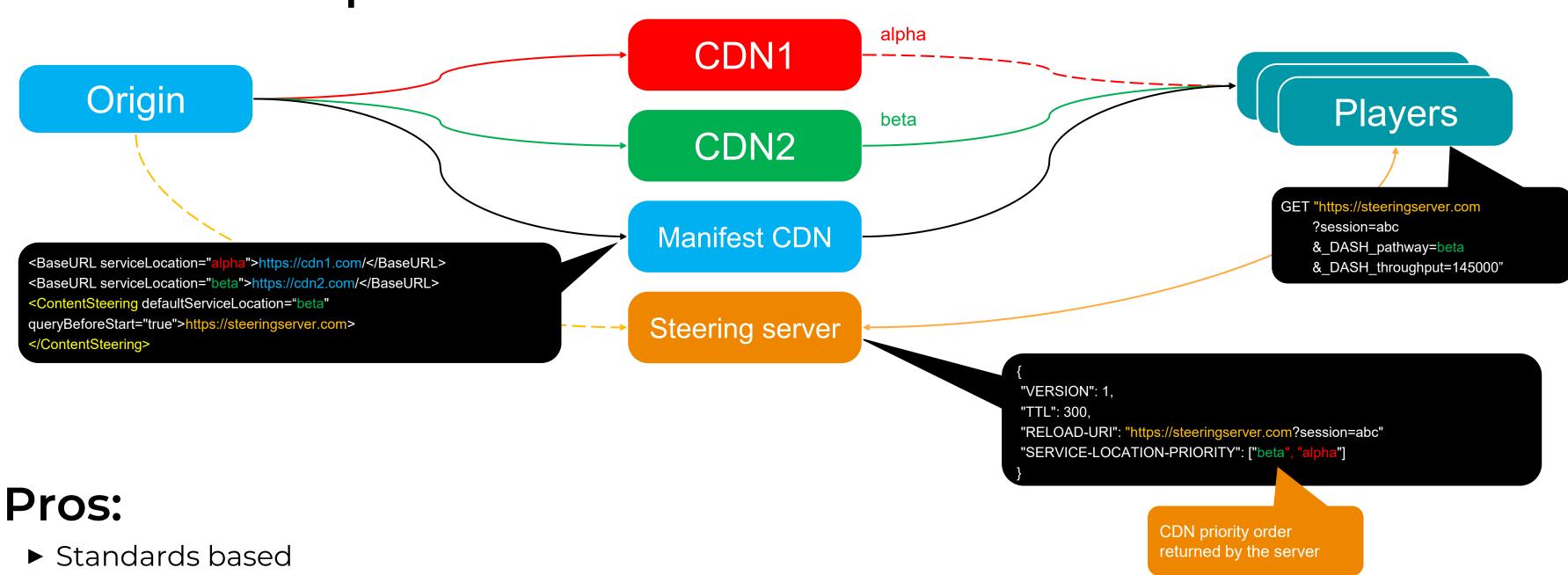
Deployment architectures

Existing clients, servers, and open-source projects



HLS/DASH CONTENT STEERING

General concept

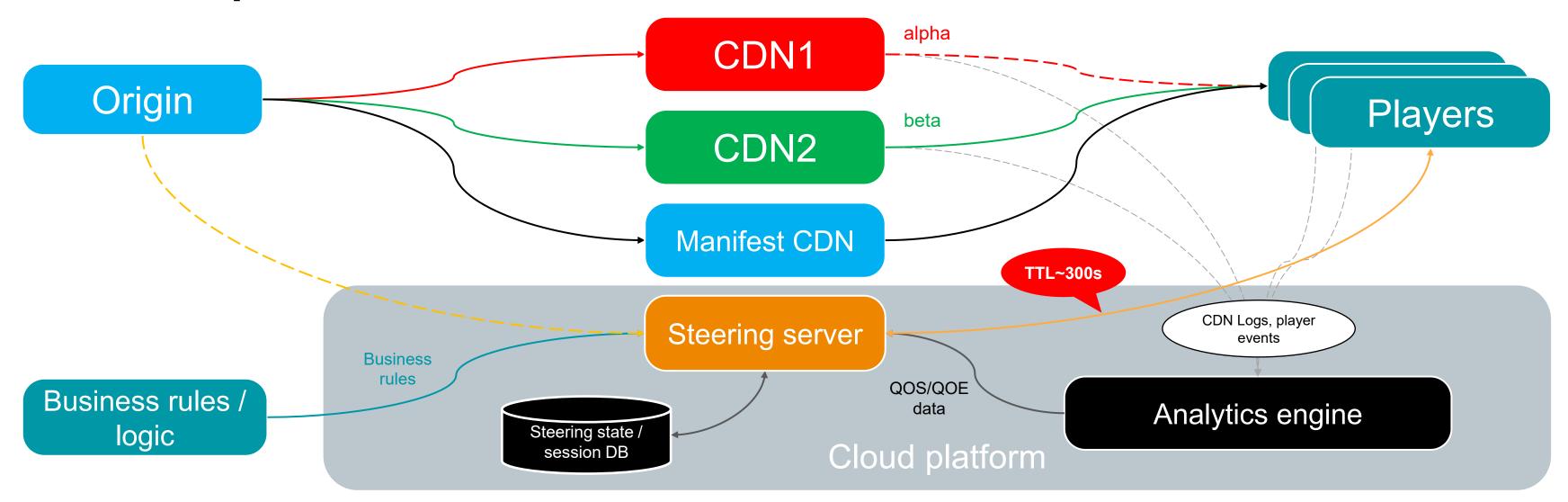


- ▶ The same steering protocol & server is used for both HLS and DASH
- ► Simple integration no need to patch players!
- ► Backwards compatible
- ► Complements the existing BaseURL redundancy / failover behavior mechanisms



CONTENT STEERING SERVER

Direct implementation



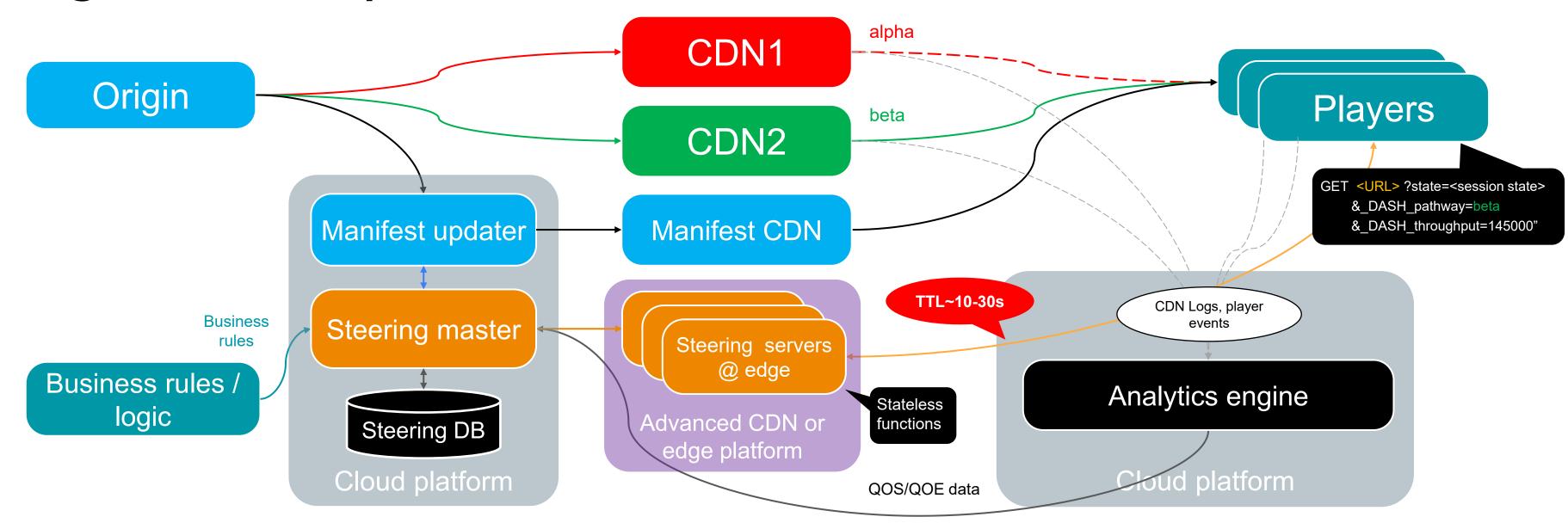
Challenges / tradeoffs:

- ▶ 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

Edge-based implementation

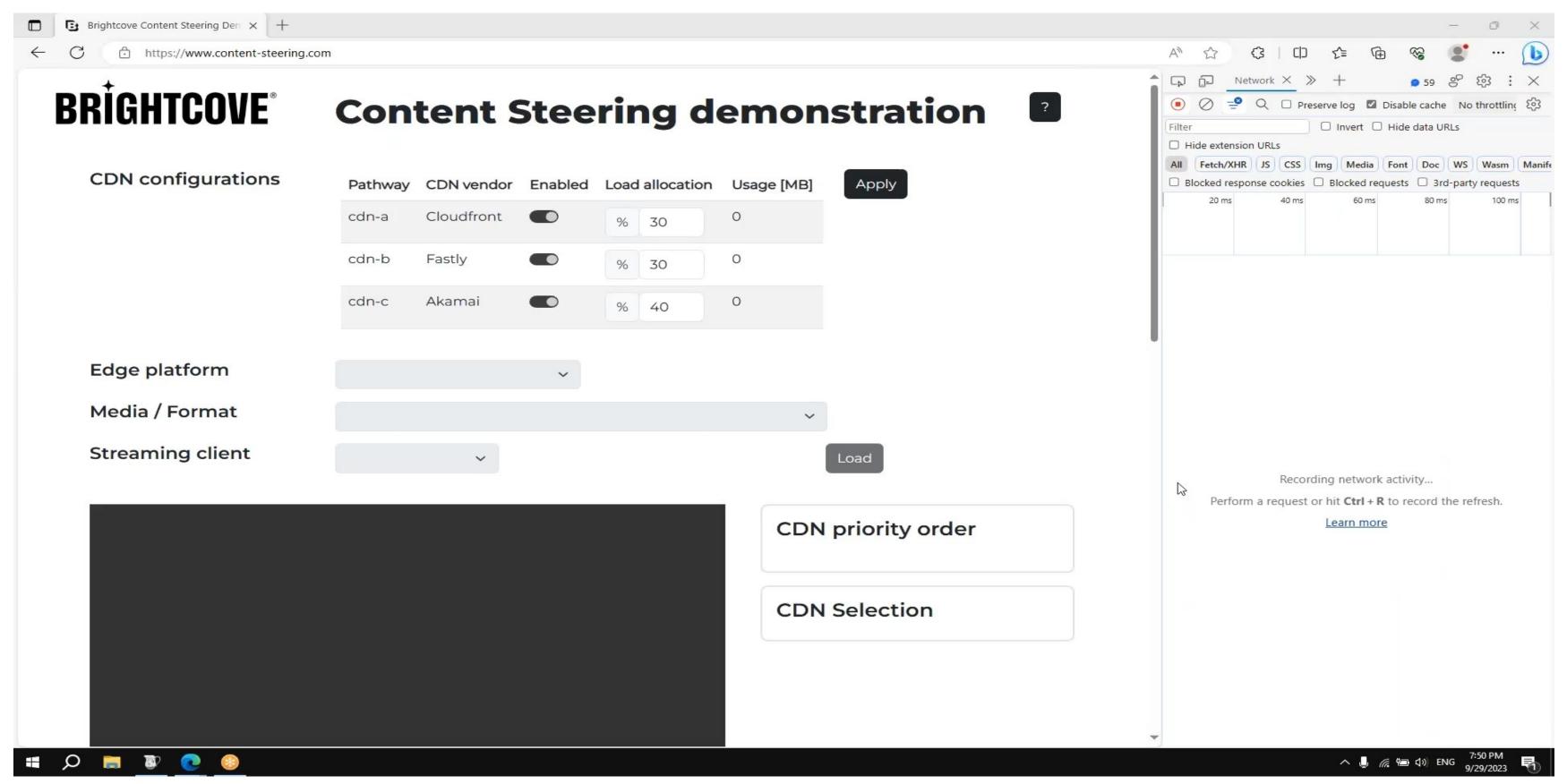


Advantages:

- ► TTL can be much smaller comparable to player buffer delay
- ► Enables QOE optimizations, faster switching / failover, more precise load-balancing.
- ► Scales well. Multiple CDNs or platforms can be used for redundancy.



TECHNOLOGY DEMONSTRATION





INDUSTRY ADOPTION

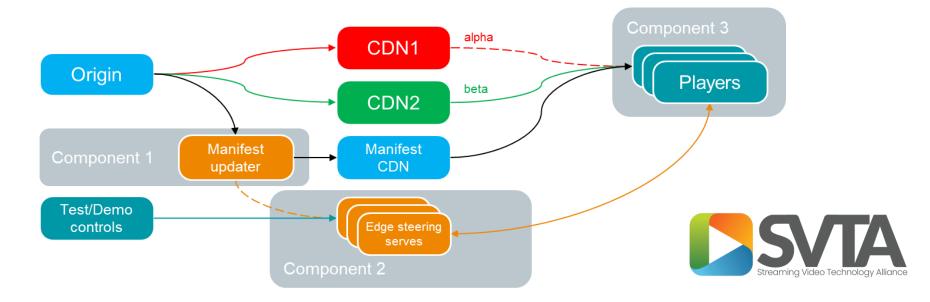
Significant momentum

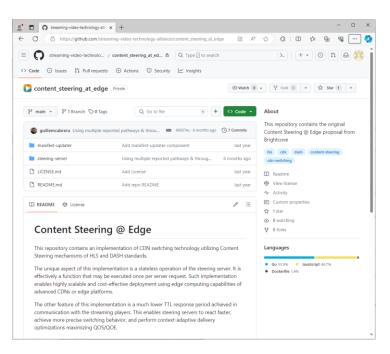
▶ Players:

Media Player	DASH Content Steering	HLS Content Steering
dash. js	since version 4.5.0	
⋘VIDEO .JS	since version 8.8.0	since version 8.8.0
Shaka _{Player}	since version 4.6.0	since version 4.6.0
G ExoPlayer	planned for 2025	planned for 2025
hls .js		since version 1.4.0
É AVPlayer		since iOS version 15.0

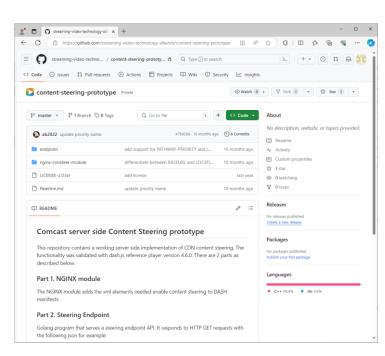
- ► Packagers & servers:
 - > Brightcove, Synamedia, Einbliq.io
 - > Shaka packager, SVTA open-source projects
- ► Standards & guidelines
 - ► IETF RFC 8216bis (HLS)
 - ► ISO/IEC 23009-1 (DASH)
 - ► ETSI TS 103 998 (DASH-IF Content Steering)

► Open-source tools





https://github.com/streaming-video-technologyalliance/content_steering_at_edge



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

STEERING ALGORITHMS

Distribution of traffic

Managing CDN costs and commits

QOS- and QOE-driven optimizations

DISTRIBUTION OF TRAFFIC

Conceptual model

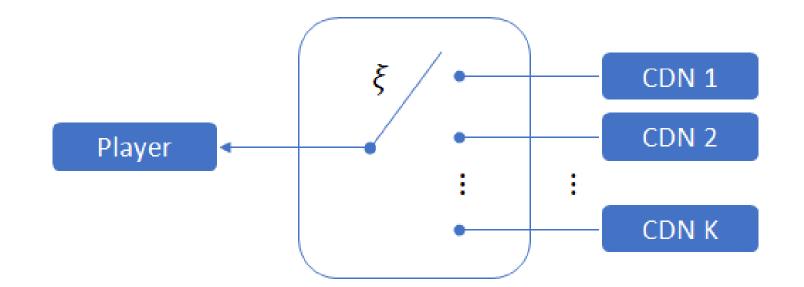
- ► A system with K CDNs
- ► Target CDN load distribution:

$$\xi = \{\xi_1, \dots, \xi_k\}, \qquad \xi_i \in [0,1], \qquad \Sigma \, \xi_i = 1.$$

► CDN selector = a generator of indices $x \in \{1, ..., K\}$, such that relative frequencies of occurrence:

$$f(x) \to \xi_x$$

▶ NB: all known science. Knuth V2.



THE CLASSIC WORK NEWLY UPDATED AND REVISED The Art of Computer Programming VOLUME 2 Seminumerical Algorithms Third Edition DONALD E. KNUTH

Generating priority lists:

- ▶ Let $x_1 \in \{1, ..., K\}$ be first CDN index retrieved by a generator for distribution $\xi^{(1)} = \xi$
- To retrieve 2nd index, we can use a modified distribution produced by excluding x_1 from $\{1, ..., K\}$ $\xi^{(2)} = \left[\xi_i^{(2)} = \xi_i^{(1)} / \left(1 \xi_{x_1}^{(1)}\right), i \in \{1, ..., k\} \setminus x_1\right]$
- ► The process is repeated recursively, until we generate a list of K CDN indices.

Realizations

- ► CDN choices can be implemented at different level of granularity when delivering the content
 - > Per asset
 - > Per session, or per manifest update period in live streaming
 - > Per TTL interval during the session.
- For long-tail content, per-asset allocation may be adequate
- ► Per-session or per-manifest update period methods are realizable with manifest rewrites
- ► For finer granularity updates one needs Content Steering standard!



MANAGING CDN COSTS AND COMMITS



MOTIVATIONAL EXAMPLE

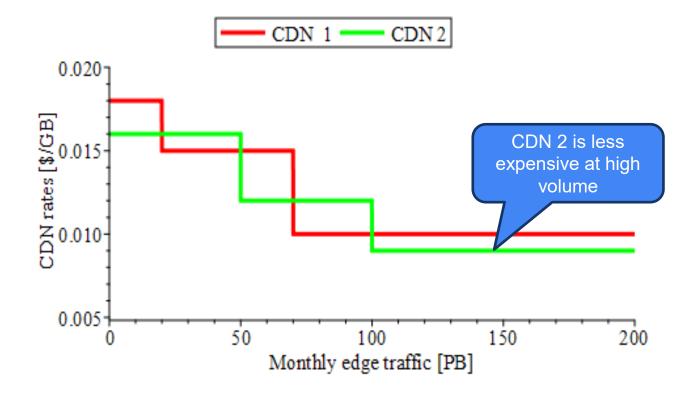
Example system:

- $ightharpoonup V_1, V_2 = edge traffic [GB/m]$
- $Arr R_1(V_1), R_2(V_2) = \text{CDN edge traffic rates [$/GB/m]}$
- ► $C(V_1, V_2) = V_1 \cdot R_1(V_1) + V_2 \cdot R_2(V_2)$ = total CDN delivery cost



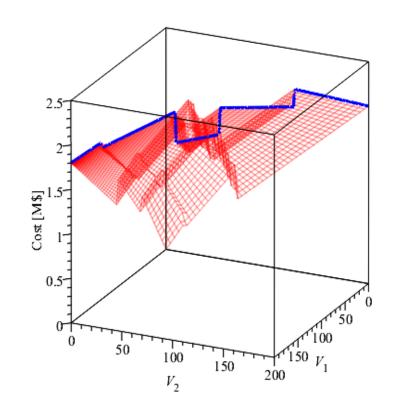
CDN rates:

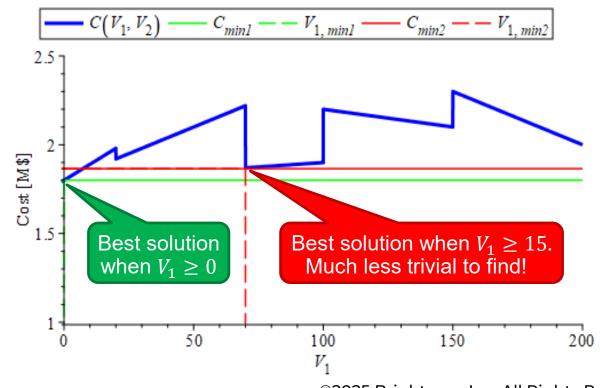
$$R_1(V_1) = \begin{bmatrix} 0.018, & V_1 < 20 \\ 0.015, & 20 \le V_1 < 70, & R_2(V_2) = \begin{bmatrix} 0.016, & V_2 < 50 \\ 0.012, & 50 \le V_2 < 100. \\ 0.009, & V_2 \ge 100 \end{bmatrix}$$



Total delivery cost when $V_1 + V_2 = 200$ [PB]:

- ▶ $V_{1,min1} = 0$ -- best solution when $V_1 \ge 0$
- ▶ $V_{1,min2} = 70$ -- best solution with commitment: $V_1 \ge 15$ [PB]



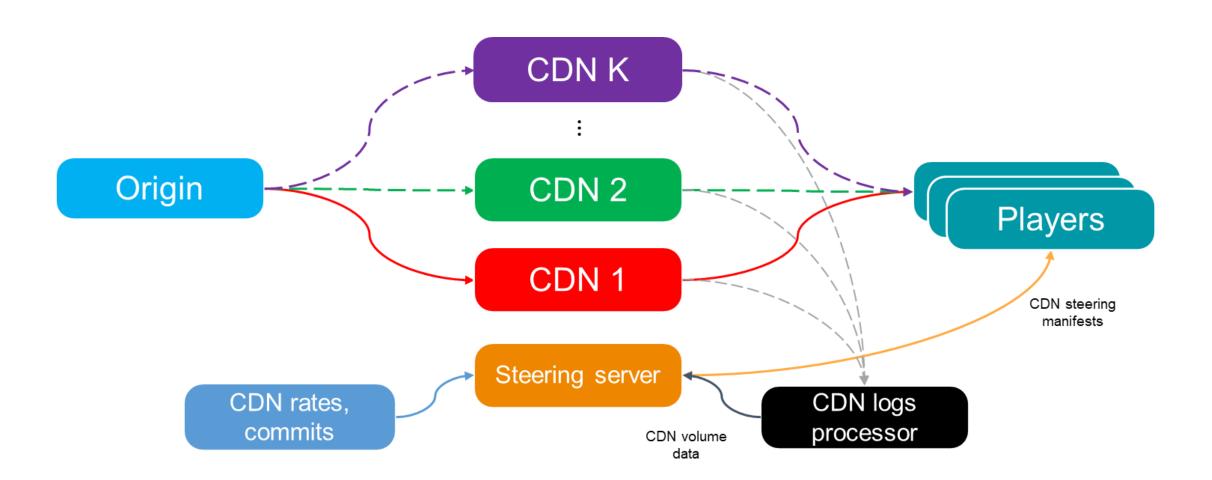




MODEL OF A DELIVERY SYSTEM

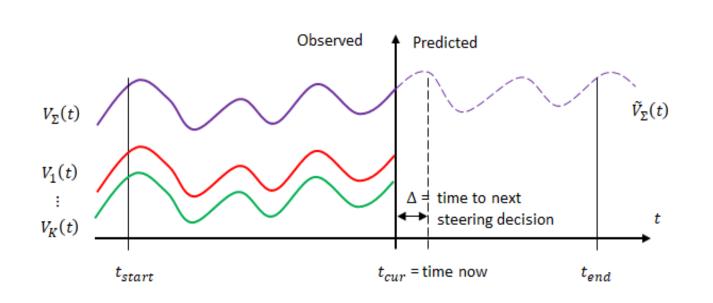
System architecture

- Content steering server tells players which CND to use
- ▶ Dynamically assigns load factors $\xi = [\xi_1, ..., \xi_K], \ \xi_i \in [0,1], \ \Sigma \xi_i = 1.$
- Uses CDN statistics and contractual data as inputs to the optimization problem



CDN edge volume statistics:

- ► K = the number of the CDNs
- $ightharpoonup V_i(t)$ = observed traffic volume at CDN i
- \blacktriangleright $V_{\Sigma}(t) = \sum_{i=1}^{K} V_{i}(t)$ = combined volume on all CDNs
- $ightharpoonup [t_{start}, t_{end}] = \text{commit period}$
- $ightharpoonup V_i^{commit}$ = minimum volume committed for CDN i



MANAGING CDN VOLUME COMMITS

Commit period

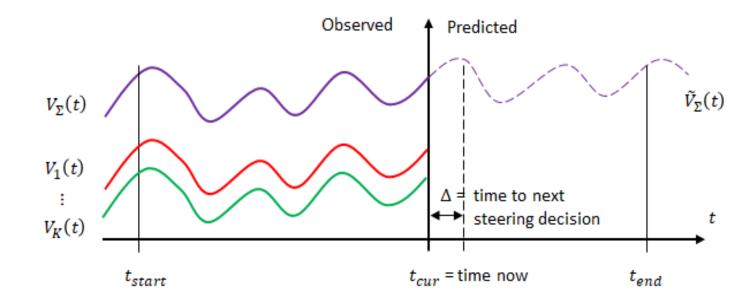
- ▶ $t \in [t_{start}, t_{end}]$ = the entire period
- ▶ $t \in [t_{start}, t_{cur}]$ = processed
- ▶ $t \in [t_{cur}, t_{end}]$ = remainder in the future

Future traffic ($t \ge t_{cur}$):

- $\qquad \qquad \tilde{V}_{\Sigma}(t) = Pred(\{V_{\Sigma}(t), t \leq t_{cur}\}, t)$
- $\blacktriangleright \ \tilde{V}_i(t) = \xi_i \cdot \tilde{V}_{\Sigma}(t)$

Total traffic in commit period

- $V_i^- = V_i(t \in [t_{start}, t_{cur}]) = \int_{t_{start}}^{t_{cur}} V_i(t) dt$
- $\tilde{V}_{\Sigma}^{+} = \tilde{V}_{\Sigma}(t \in [t_{cur}, t_{end}]) = \int_{t_{cur}}^{t_{end}} \tilde{V}_{\Sigma}(t) dt$
- $\qquad \qquad \tilde{V}_i^{total} = V_i^- + \xi_i \cdot \tilde{V}_{\Sigma}^+$



- = predicted traffic for all CDNs
- = load factors assigned to CDNs for the remainder of the period
- = predicted traffic for i-th CDN
- = traffic delivered by i-th CDN in the past
- = predicted total traffic for all CDNs till the end of commit period
- = predicted total traffic for i-th CDN in the commit period

CDN volume commitments:

 $\blacktriangleright \ \ V_i^{total} \geq V_i^{commit} \ \Rightarrow \ \ \tilde{V}_i^{total} \geq V_i^{commit} \ \Rightarrow$

$$\xi_i \ge \frac{V_i^{commit} - V_i^-}{\widetilde{V}_{\Sigma}^+}$$

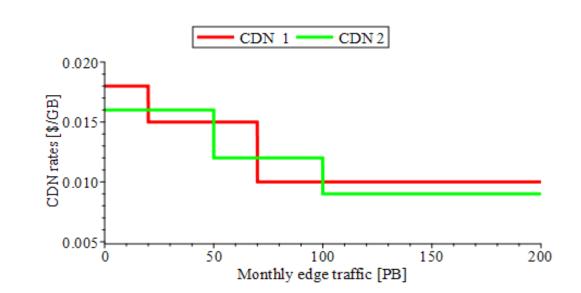
Expressed as limits for CDN load factors



CDN DELIVERY COSTS

CDN edge traffic rates:

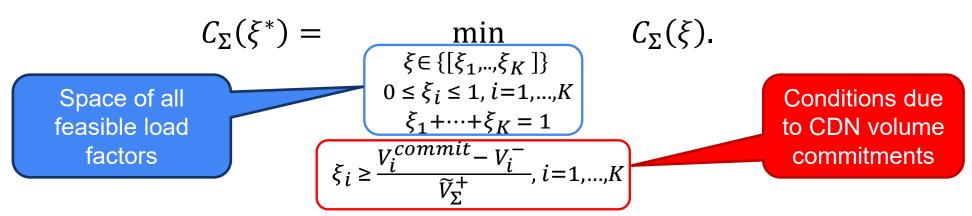
$$R_{i}(V) = \begin{bmatrix} R_{i}^{1} & if & V < V^{1} \\ R_{i}^{2} & if & V^{1} \leq V < V^{2} \\ \vdots & \vdots & \vdots \\ R_{i}^{T} & if & V^{T-1} \leq V \end{bmatrix}, i = 1, \dots, K$$



Overall delivery cost:

$$C_{\Sigma}(\xi) = \sum_{i=1}^{K} \tilde{V}_{i}^{total} R_{i} (\tilde{V}_{i}^{total}) = \sum_{i=1}^{K} (V_{i}^{-} + \xi_{i} \cdot \tilde{V}_{\Sigma}^{+}) \cdot R_{i} (V_{i}^{-} + \xi_{i} \cdot \tilde{V}_{\Sigma}^{+}).$$

Optimization problem:



The challenge

• $C_{\Sigma}(\xi)$ is discontinuous! Not clear if this problem is solvable by the existing techniques.

SOLVING THE OPTIMIZATION PROBLEM

Let's introduce a space of CDN pricing tiers:

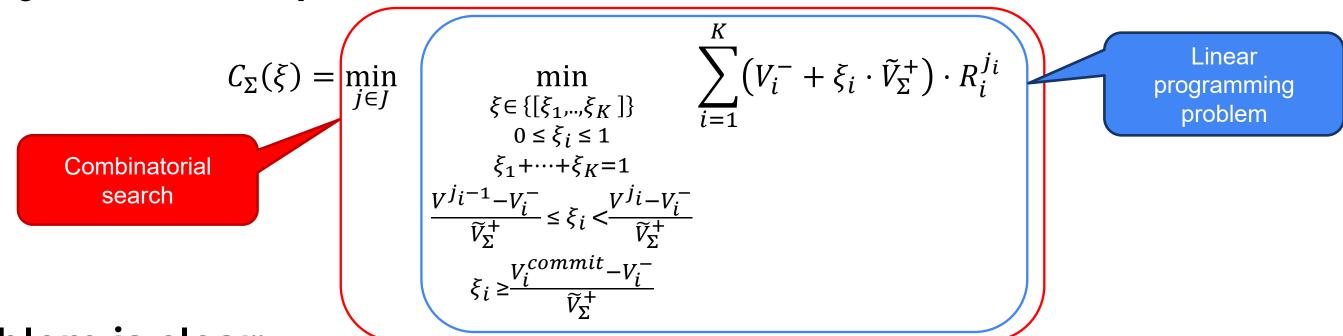
$$J = \{[j_1, \dots, j_K], j_i \in \{1, \dots, T-1\}, i = 1, \dots, K\}, \qquad |J| = T^K$$

If we know $j \in J$, then:

$$C_{\Sigma}(\xi) = \sum_{i=1}^{K} \left(V_{i}^{-} + \xi_{i} \cdot \tilde{V}_{\Sigma}^{+} \right) \cdot R_{i} \left(V_{i}^{-} + \xi_{i} \cdot \tilde{V}_{\Sigma}^{+} \right) = \sum_{i=1}^{K} \left(V_{i}^{-} + \xi_{i} \cdot \tilde{V}_{\Sigma}^{+} \right) \cdot R_{i}^{j_{i}}$$

$$V^{j_{i}-1} \leq \tilde{V}_{i}^{total} < V^{j_{i}} \implies V^{j_{i}-1} \leq V_{i}^{-} + \xi_{i} \cdot \tilde{V}_{\Sigma}^{+} < V^{j_{i}} \implies \frac{V^{j_{i}-1} - V_{i}^{-}}{\tilde{V}_{\Sigma}^{+}} \leq \xi_{i} < \frac{V^{j_{i}} - V_{i}^{-}}{\tilde{V}_{\Sigma}^{+}}, \qquad i = 1, \dots, K$$

Consequently, the entire problem reduces to:



Now the problem is clear:

- Superposition or combinatorial enumeration and linear programming problems!
- Solvable by using the existing methods.

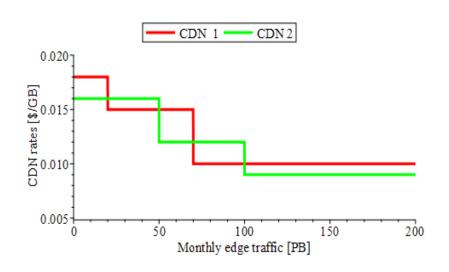


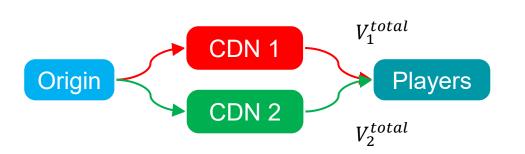
EXAMPLE SOLUTION

CDN rates:

$$> j(V) = \begin{bmatrix} 1, & V^0 \le V < V^1 \\ 2, & V^1 \le V < V^2 \\ 3, & V^2 \le V < V^3 \\ 4, & V^3 \le V < V^4 \\ 5, & V^4 \le V < V^5 \end{bmatrix} \begin{bmatrix} 0.018, & j = 1 \\ 0.015, & j = 2 \\ 0.015, & j = 3 \\ 0.010, & j = 4 \\ 0.010, & j = 5 \end{bmatrix} R_2(j) = \begin{bmatrix} 0.016, & j = 1 \\ 0.016, & j = 2 \\ 0.012, & j = 3 \\ 0.012, & j = 4 \\ 0.009, & j = 5 \end{bmatrix}$$

> Volume limits: $V^j = [0, 20, 50, 70, 100, ∞]$





Predicted total traffic and commits:

- $V_{\Sigma}^{total} = 200 \text{ [PB]}$
- $> V_1^{commit} = V_2^{commit} = 15 [PB]$
- $V_1^{total} = \xi_1 \cdot V_{\Sigma}^{total}$, $V_2^{total} = \xi_2 \cdot V_{\Sigma}^{total}$, $\xi_1 = \text{load factor for } 1^{\text{st}} \text{ CDN}$, $\xi_2 = 1 \xi_1 = \text{load factor for } 2^{\text{nd}} \text{ CDN}$.

Combinatorial space:

>
$$5^2 = 25$$
 combinations of pairs: $[j_1, j_2]$. E.g., $[j_1 = 4, j_2 = 5] \Rightarrow \begin{cases} 70 \le V_1^{total} < 100 \\ 100 \le V_2^{total} < \infty \end{cases}$

LP problems:

$$\begin{split} C_{\Sigma}(\xi | \ [j_{1}, j_{2}]) &= \min_{\substack{\xi_{1} \in [0, 1] \\ \frac{V^{j_{1} - 1}}{V_{\Sigma}^{total}} \leq \xi_{1} < \frac{V^{j_{1}}}{V_{\Sigma}^{total}}, \ \frac{V^{j_{2} - 1}}{V_{\Sigma}^{total}} \leq 1 - \xi_{1} < \frac{V^{j_{2}}}{V_{\Sigma}^{total}}} \\ & \xi_{1} \geq \frac{V_{1}^{commit}}{V_{\Sigma}^{total}}, \ 1 - \xi_{1} \geq \frac{V_{2}^{commit}}{V_{\Sigma}^{total}} \end{split}$$

Example solution:

$$C_{\Sigma}(\xi | [j_{1} = 4, j_{2} = 5]) = \min_{\substack{0.35 \leq \xi_{1} < 0.5, \, 0.5 \leq 1 - \xi_{1} < \infty \\ \xi_{1} \geq 0.075, \, 1 - \xi_{1} \geq 0.075}} 200 \cdot (\xi_{1} \cdot 0.01 + (1 - \xi_{1}) \cdot 0.009) \Rightarrow \xi_{1} = 0.35, \qquad V_{1}^{total} = 70, \qquad C_{\Sigma} = 1.87.$$



MULTI-REGIONAL QOE-BASED OPTIMIZATIONS



CONCEPTUAL MODEL

System architecture

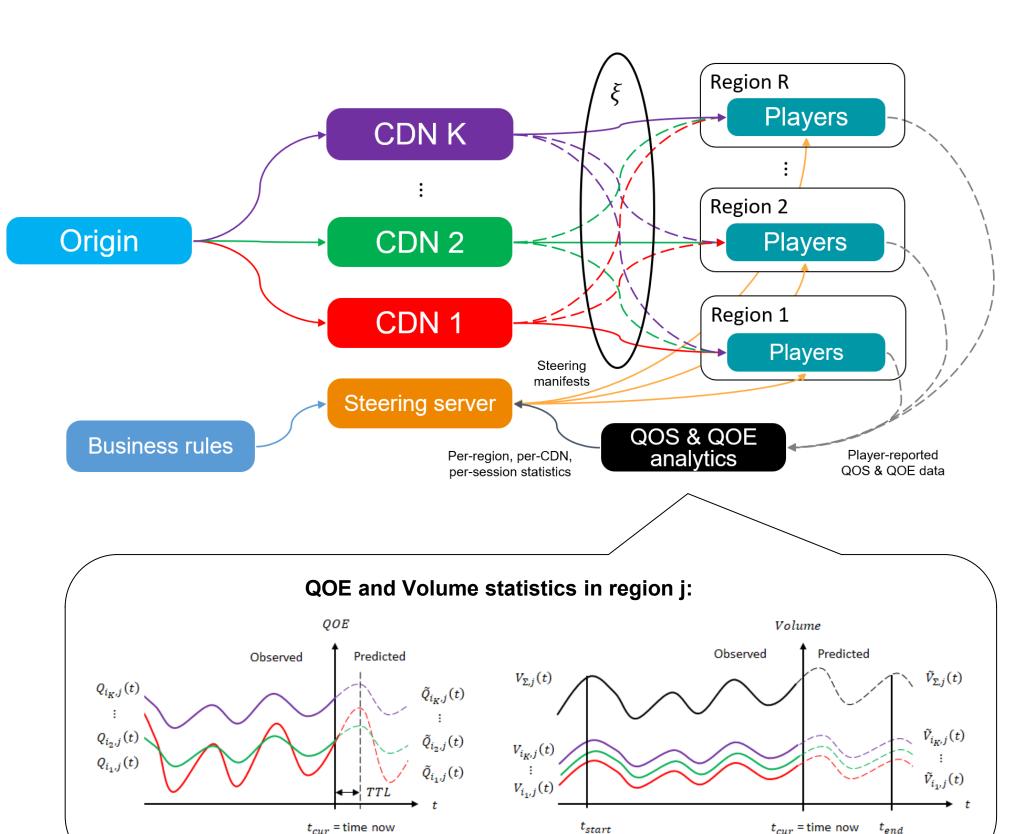
- ► System with K CDNs, delivering content to R regions
- ► Content steering server maintains a matrix of distribution of CDN loads across the regions:

$$\xi = \begin{bmatrix} \xi_{11} & \cdots & \xi_{1R} \\ \vdots & \ddots & \vdots \\ \xi_{K1} & \cdots & \xi_{KR} \end{bmatrix}, \qquad \xi_{ij} \in [0,1], \qquad \sum_{i=1}^K \sum_{j=1}^R \xi_{ij} = 1$$

- ▶ Performs per-asset/session/TTL-level CDN selection to shape CDN edge volume $\rightarrow \xi$
- ► Uses volume and QOS/QOE statistics and business rules as inputs to optimize E for a particular objective

QOE and edge volume statistics:

- $ightharpoonup Q_{ij}(t)$ = observed quality for CDN i in region j
- $ightharpoonup V_{ii}(t)$ = edge volume at CDN i in region j
- $V_{\Sigma,j}(t) = \sum_{i=1}^{K} V_{ij}(t)$ = combined volume in region j
- $V_{i,\Sigma}(t) = \sum_{j=1}^{R} V_{ij}(t)$ = global volume delivered by CDN i
- ightharpoonup [t_{start} , t_{end}] = commit period
- V_i^{commit} = minimum volume committed for CDN i



MULTI-REGIONAL QOE OPTIMIZATION

Variables

- $ightharpoonup Q_{ij}(t)$ = observed quality for CDN i in region j
- Average quality

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

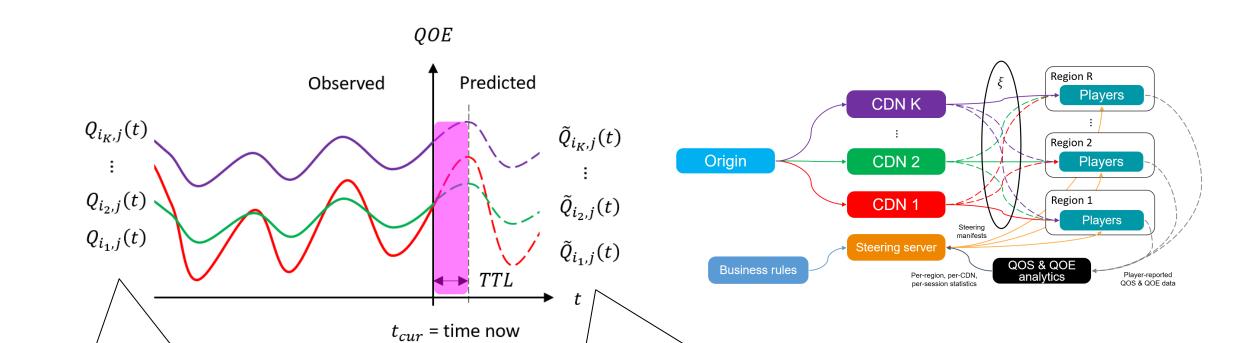
Average quality during the next TTL interval:

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

CDN load factors:

$$\boldsymbol{\xi} \in \boldsymbol{\Xi} = \left\{ \begin{bmatrix} \boldsymbol{\xi}_{11} & \cdots & \boldsymbol{\xi}_{1R} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\xi}_{K1} & \cdots & \boldsymbol{\xi}_{KR} \end{bmatrix}, \boldsymbol{\xi}_{ij} \in [0,1], \boldsymbol{\Sigma}_{ij} \boldsymbol{\xi}_{ij} = 1 \right\}$$

 $ightharpoonup \zeta_i = \sum_{j=1}^R \xi_{ij}$ = load on CDN i



 $Q_{ij}(t)$ = any suitable QOE metric of fused combination of such metrics as reported by the system.

E.g. it can be a function of (a) startup time, (b) buffering rate, (c) average delivered resolution, (d) average delivered encoding quality, etc.

 $\tilde{Q}_{ij}(t)$ = predicted QOE metrics in the future. Such prediction is necessary to make best decisions in the next steering period (TTL interval) Predictions can be computed using linear predictors with a combination of both short term (last hour) and long term (past day, week, same day last year) samples.

Optimization problems

Best quality under fixed global CDN load allocation:

$$\bar{Q}(\xi^*) = \min_{\xi \in \Xi} \bar{Q}(\xi).$$

$$\sum_{j=1}^{L} \xi_{ij} = \zeta_{target,i}, i=1,...,K$$

Classic linear programming problem!

▶ Best quality under constrained global CDN load allocation:

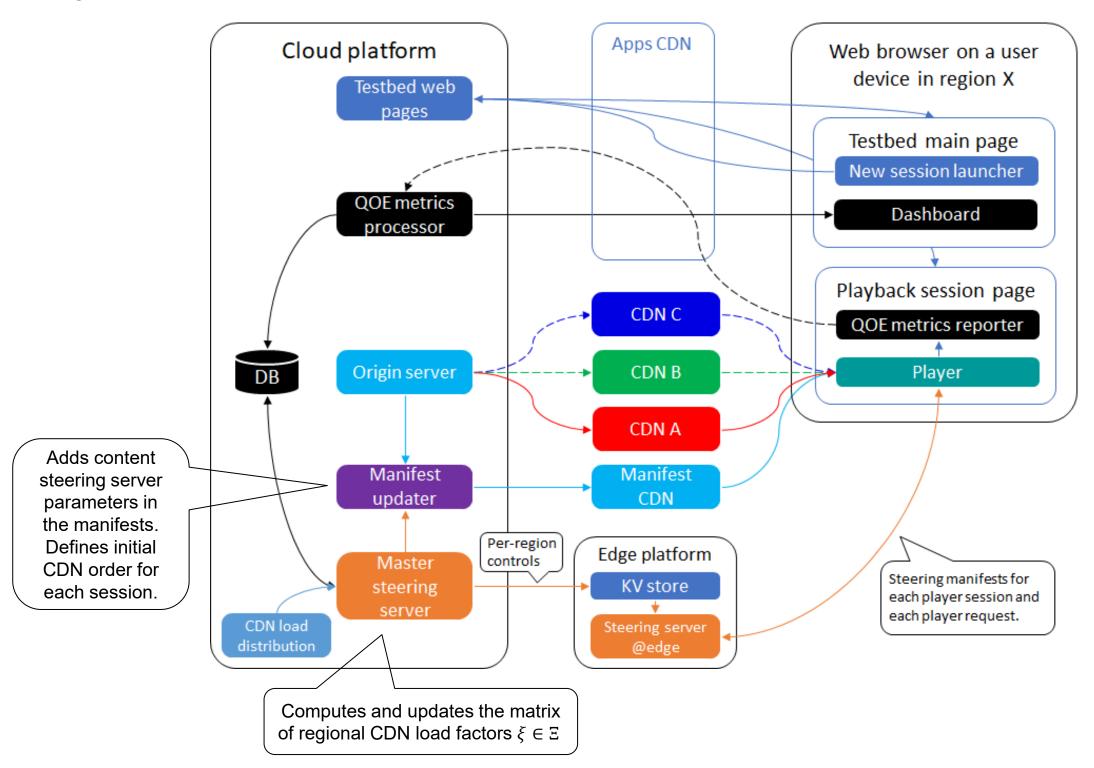
$$\bar{Q}(\xi^*) = \min_{\substack{\xi \in \Xi \\ \sum_{j=1}^L \xi_{ij} \ge \zeta_{min,i}, i=1,\dots,K}} \bar{Q}(\xi).$$

Classic linear programming problem!

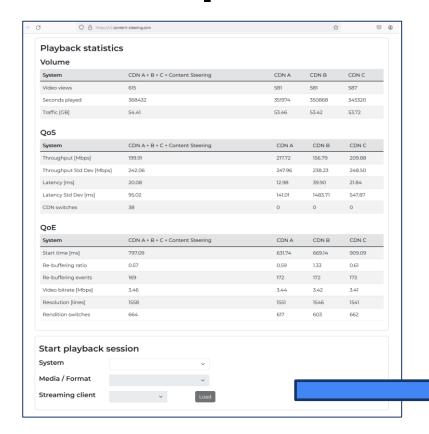


EXPERIMENTAL FRAMEWORK

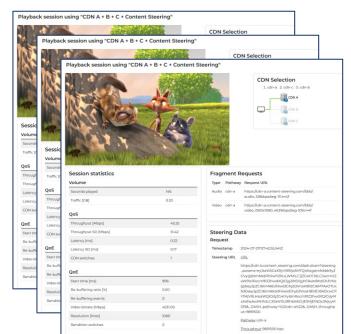
System architecture



Control panel:



Playback sessions:



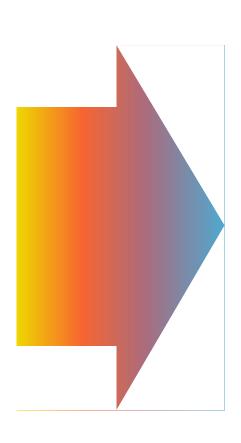




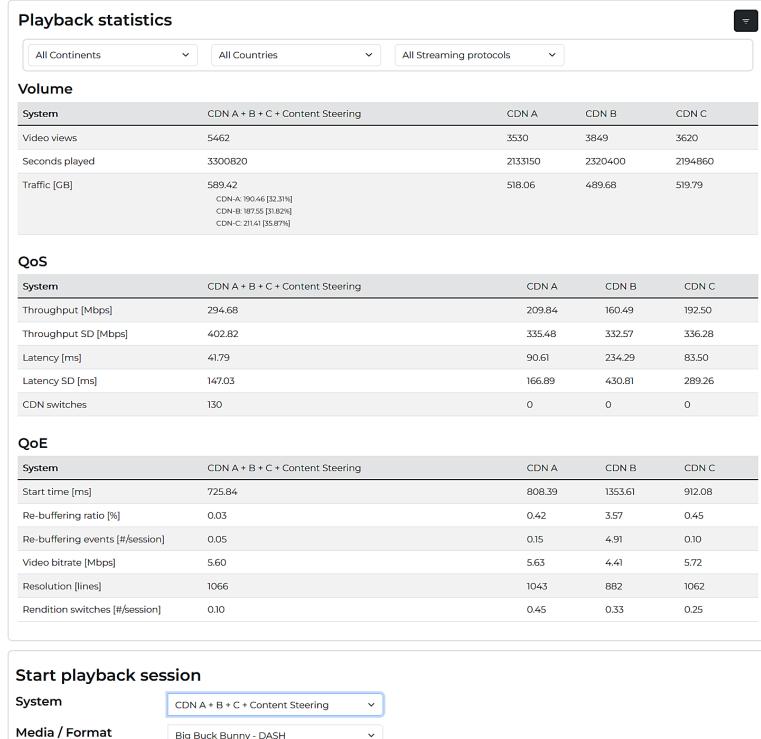
PERFORMANCE STUDY

Methodology

- Systems compared:
 - > CDN A/B/C single-CDN systems
 - > CDN A+B+C multi-CDN system with content steering
- Testing orchestration
 - > Over 150 proxies utilized
 - Broad set of geo locations
 - > Using K6, and several VPN networks
- Volumes
 - > Over 15000 sessions
 - > Over 2000 of playback hours
 - > Over 2.1TB of media data delivered
- Protocols & clients:
 - DASH, DASH.js client
 - > HLS, HLS.js client
- ► Steering servers, analytics processing
 - Brightcove



www.testbed.content-steering.com





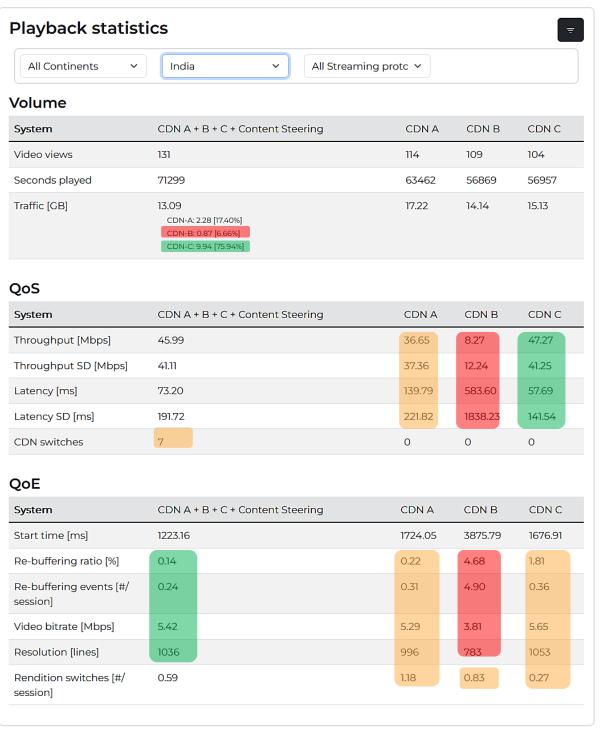
PERFORMANCE STUDY

Per-regional load adaptations

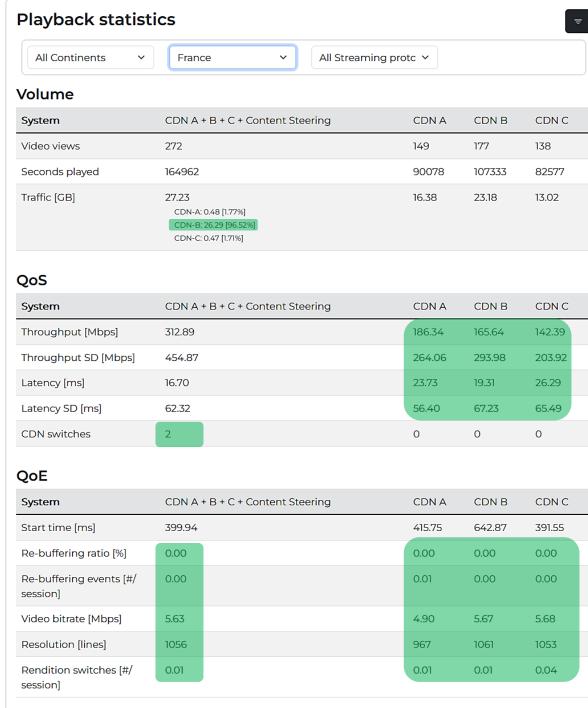
Observations

- ► Performance of CDNs can be very different across different regions
- ► In some regions the differences between CDN performance can be significant.
 - E.g. in India, we see that CDN B is struggling (much lower bandwidth and higher latency).
 - In such cases, we note that our optimizer moves traffic away from underperforming CDNs.
- ► In some other regions, we see that performance of all CDNs maybe good enough
 - > E.g. in France (in our sample), all CDNs delivered over 140Mbps in bandwidth
 - > In such regions, we move more traffic to CDNs underperforming in other locations (we move 96% of traffic to CDN-B in this region).
- ► This exercise proves that as a concept, multi-regional load allocation works!

India



France

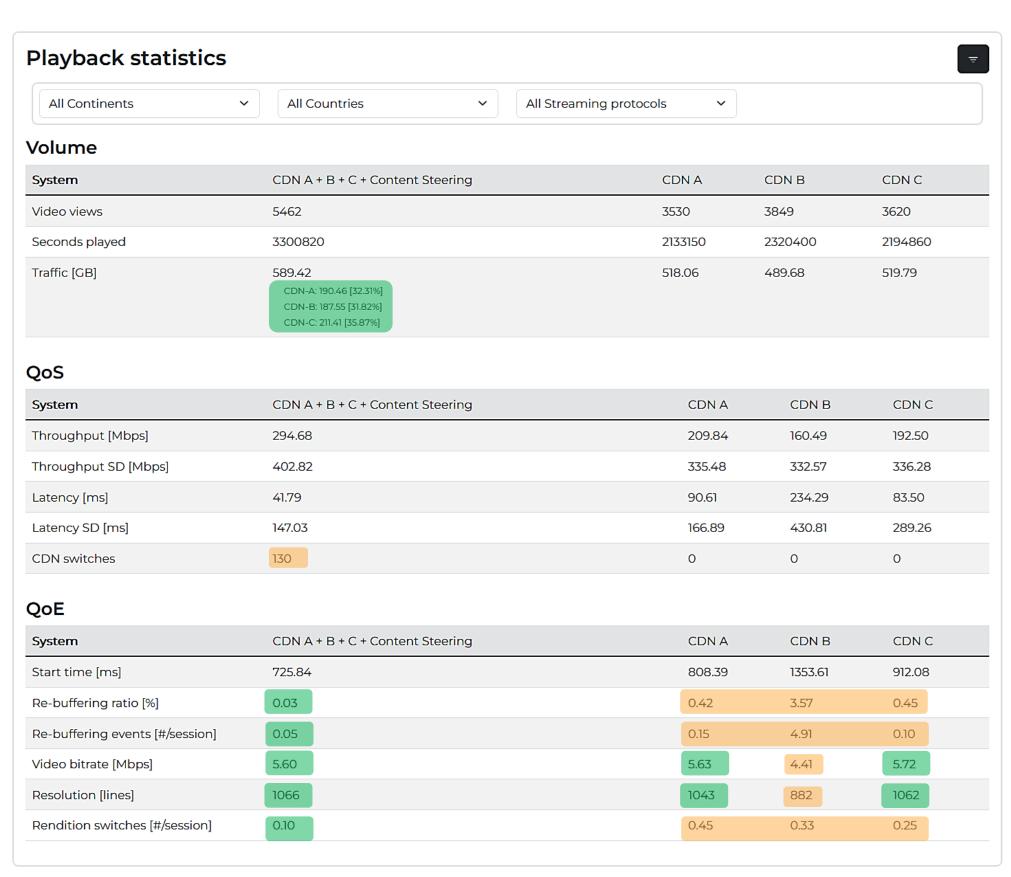


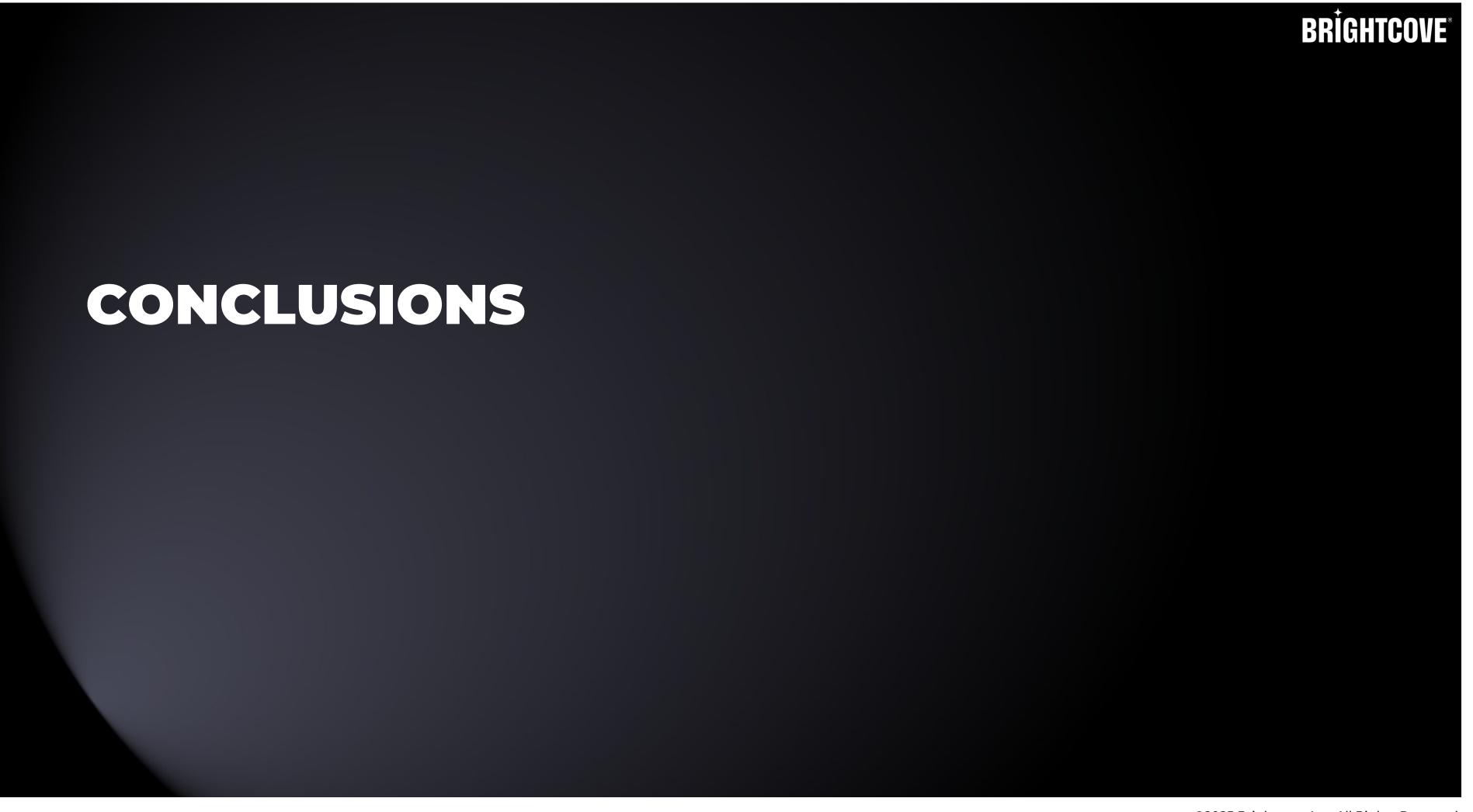


PERFORMANCE STUDY

Overall effects

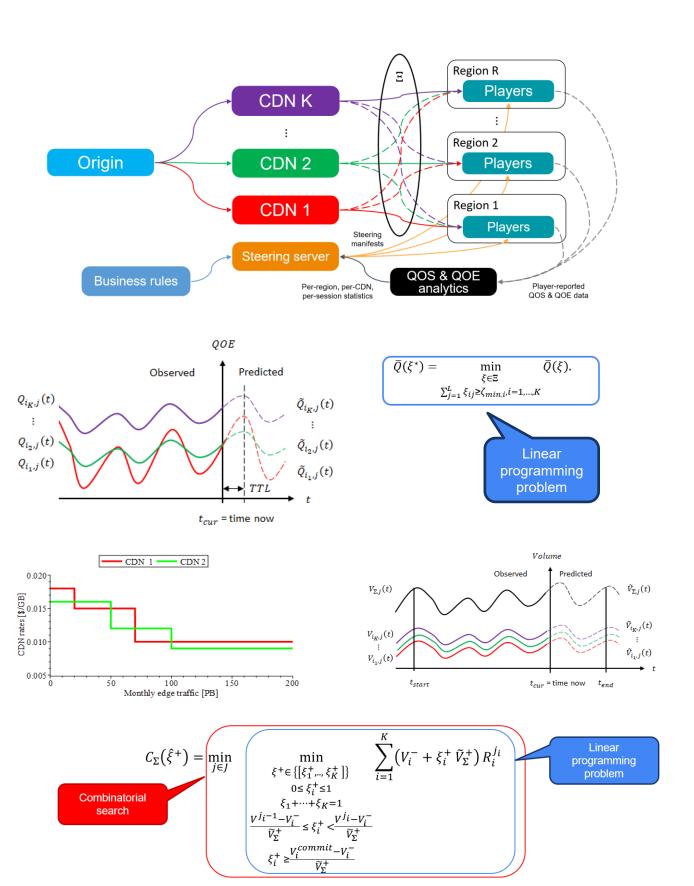
- ► Global-scale CDN loads are nearly uniform as was the optimization constraint in the optimizer
- ► In our tests sample of 5.4K sessions, the system executed 130 in-session switches – the cases when CDN performance was dropping significantly, warranting in-session switch
- ► The system with adaptive CDN switching has also achieved some notable improvements in overall QOE metrics:
 - > Lower buffering ratio
 - > Lower number of buffering events
 - > Fewer rendition switches
- The befits of adaptive CDN switching are evident!





CONCLUSIONS

- ► The design of multi-CDN systems is not trivial, but, fundamentally, it is tractable by using the existing methods and techniques. It is definitely not a "black art".
- The HLS/DASH content steering standard reduces the complexity of implementing seamless switching. It does it in a backward-compatible fashion, enabling incremental roll out of services based on this standard.
- ► There are already many clients and content publishing tools supporting this standard.
- ► Increasing support of CMCD metadata closes the gap on the analytics gathering methods.
- ► The design of steering servers is more fun, but such technologies are now being actively developed by many vendors. Some vendors will offer them as part of turn-key solutions. Some others may offer them as open-source.
- ▶ Overall, considering all the recent progress, the use multi-CDN solutions for streaming is now poised to become more common. Powered by content steering and CMCD standards, as well as proven optimization methods and strategies, such solutions are promising to be simpler, more reliable, and easier to operate and deploy.



REFERENCES Related publications

- Y. Reznik, G. Cabrera, D. Silhavy, S. Pham, A. Giladi, A. Balk, A. C. Begen, and W. Law, SMPTE Motion Imaging Journal, Jan/Feb 2025.
- 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," IBC 2024, Amsterdam, NL, September 13-16, 2024.
- Y. Reznik, G. Cabrera, On Multi-CDN Delivery Costs Optimization Problem, IEEE ISM'04, Tokyo, JP, Dec. 11-13, 2024.
- 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.
- 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).
- 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, IEEE International Conference on Fog and Edge Computing (ICFEC 2024), Philadelphia, PA, USA, May 06-09, 2024.
- 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, February 11-14, 2024.
- D. Silhavy, S. Pham, A. Giladi, A. Balk, A. Begen, and W. Law, Real-Time Streaming Reliability and Performance Optimization Using Content Steering, SMPTE Media Technology Summit, Hollywood, CA, October 16-19, 2023.
- 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, IBC 2023, September 15-18, 2023.

ECHNICAL PAPER

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



Content Steering: A Standard for Multi-CDN Streaming

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.

SMPTE MOTION IMAGING JOURNAL | JANUARY/FEBRUARY 2025

1545-0279/25@2025SMPTE

Abstract
Content Steering for DASH (ETSI TS 103 998) is a n standard developed by the DASH Industry For (DASH-I) (additional properties of the properties of the properties of the service of the service

s well known, most videos sent over the internet are delivered using streaming technologies. "The two most commonly used streaming protocols today are HTTP Live Streaming (HS) and Dynamic Adaptive Streaming over HTTP (IOASH). Both are international standards. Both use HTTP as the underlying network protocol and employ content delivers models. (TMS) for Alexton total N.

is simple: the media content is encoded, packaged, a placed on the origin server first. CDN then propagates through its system of caches, to a vast and geographic dispersed population of viewers. Effectively, CDN enab mass-scale delivery. (50)

able in all relevant regions some may have a saturated it ternal network, and some may not have sufficient capacit of edge caches to support the delivery of a vest and divers catalog of media content. Occasionally, CDNs may als experience outages or technical failures, making them un available for some time.^{5,0}

1545-0279/25©2025SMPTE

Given such limits, large streaming operators increasing employ multiple CDNs and so called "CDN switching" te nologies as part of their delivery architectures. ³⁰ By inte gently distributing traffic across multiple CDNs intelligen such systems can achieve better reliability, scale, and qual of experience (QoBi delivered to end users. However, deoping and operating such multi-CDN systems are not trit tasks. ^{20,11}

Even a basic traffic switching operation between diffeent CDNs is not exactly straightforward. **Table 1** lists seve all methods and approaches tried in the past. As easily of served, none of these approaches is perfect. Each has various constant prost services and prost services and prost services.

Fortunately, this problem has been recently addressed standards.⁵³ The latest versions of HLS and DASH cifications.⁵⁶ now include vontent steering functions signed specifically for this purpose. Using these functions, et design of multi-CDN delivery systems becomes much restraightforward. Much less effort and fewer changes required across the streaming workflow. It also ensures sper and consistent switching behavior for all players immented according to HLS and DASH standards. Once deved, it is also guaranteed to work with subsequent system proponent upgrades (players, peaclagers, etc.). The design

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

The Standard

The initial concept of content-steering technology for HL
was developed by Apple in February 2021²¹. It was added to
IETF RFC 8216bis, "HTTP Live Streaming 2nd Edition," in No

squently, in July 2022, the DASH Industry Forum F) produced a technology proposal titled "Content for DASH." The DASH-IF proposal has extended content steering concept by preserving the syntax

NUARY/FEBRUARY 2025 | SMPTE MOTION IMAGING JOURNAL



Multi-Regiona using HLS benefits, challenges, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system (highers, how to address them and the capabilities of the system).

hilling between different CDNs.

Authority of design of ofderived different control of the con

Kywords—Strousling, multi-CDV; programming, DASH, HLS, content sto programming, DASH, HLS, content sto programming, DASH, HLS, content steering [1-5] is a new ex DASH [7] streaming standards, eas CDNs for delivery, Fig. 1 illustrates of this technology. In addition to the elements, such as origin servers, CDN introduces a new network entity—a ci server communicates with the ciberts which CDNs to me. The TTL parameter II.

which CDNs to me. The TLT, purmature as east serve response controls the frequency of verave-liner cloudnessys; to OCCA.

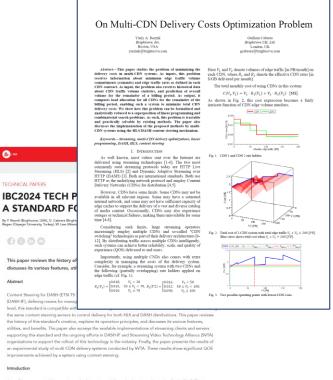
One natural utility of content sterning technology in Colcerning, if the performance of any of the CDNs degrades for example, if the performance of any of the CDNs degrades for security in the CDNs degrades of the server can room traffic any of sens it Horseever, the design of OCC optimization generally depends on the TLT purmature. The shorter it is, the faster the server can seast and deliver accounty with deficience (e.g., in prevent or death from running would significantly increase the load on the steering server. It will also increase the operating colon. Hence, one must figure and also becomes the operating colon. Hence, one must figure

that a role is an unique of an experiment of the property of

II. FORMALIZATION OF A PROBLEM
Consider a multi-CDN system with R CDNs and Roperating regions. Such regions on the identified, e_R , by
county codes, sub-divisions, 18P ASN numbers, etc.
(DN (i) and each region (ii), we can observe chains of "quality
scores" $Q_1(t)$ reported over time (i): $Q_1(t), (t = 1, ..., R, -1, ..., R)$ (i) shoth scores may represent a specific QOS or QOE metric
(e.g., buffer raish) or a fixed combinator of several metrics
(e.g., wrenge bundwidth, latency, stemp time, buffering time,

(e.g., average bondwidth, latency, startop time, buffering time,

Outpoint



©2025 Brightcove Inc. All Rights Reserved.

THARK YOU