Content Steering for HLS & DASH

draft-pantos-content-steering

Roger Pantos, Streaming Media Engineer
Apple Inc.

Challenge:

Enable streaming providers to control traffic choices

HLS refresher

Delivery landscape

Before Content Steering

Objectives

How Content Steering works

Outcomes

Content Steering spec

HTTP Live Streaming refresher

Series of short media segments

Playlist: list of segment URLs

For live, players reload Playlists

HTTP Live Streaming refresher

- Series of short media segments
- Playlist: list of segment URLs
- For live, players reload Playlists
- Top-level Playlist contains
  - Playlists for different qualities (ABR)
  - Playlists for different renditions

Delivery landscape

Playlists and segments delivered via HTTP

Scale via proxy caches (CDNs)

Many providers use multiple CDNs

- Geography / footprint
- Cost

Additional resilience

- Primary / backup encoders, origins, data centers

Dynamic delivery landscape

Preferred mix may shift by time of day

Asset transitions

Outages

Service must be uninterrupted

Before Content Steering

Servers could provide multiple redundant Playlists, but

- No control over client choice
- No control over client response to outages
- Options were static

Different players make different choices

- Even from version to version

Objectives of Content Steering

Give service providers positive control over delivery

- Both planned and unplanned

Allow adjustments during playback

Simple to implement and integrate with existing systems

Abstract core principle out of HLS

- Enable reuse and interoperability

How Content Steering works

Each Playlist assigned to a "Pathway"

Each Pathway has complete set of media

Client gets "Steering Manifest" from server

Steering Manifest has ordered list of Pathways

Client switches to highest unpenalized Pathway

Steering manifest reload

Steering Manifest contains TTL and Reload URL

Client reloads manifest after TTL expires

- New TTL + reload URL
- Updated Pathway priority list
- New Pathway clones

```
{
        "TITLE": "Content Steering Manifest Example"
    "VERSION": 1,
    "PATHWAY-PRIORITY": [ "PATH-A", "PATH-B" ]
    "TTL": 20,
    "RELOAD-URI": "./steering-manifest.json?s_weight=10",
}
```

Defined client behavior

Start on Pathway specified by Steering Manifest

If attempts to use Pathway fail, penalize Pathway

Switch to next-priority Pathway

Reload manifest according to TTL

- Retry if HTTP 429 Too Many Requests

Reevaluate after manifest update

Server-managed state

Servers put per-client state in query parameters of Reload URL

State echoed back when client reloads manifest

Client-provided state

Clients provide certain attributes to server

Added as query parameters to reload URL

- Client current Pathway
- Pathway throughput

Pathway cloning

Client starts with static set of Pathways

Steering manifest can define Pathway Clones

- Copy of existing Pathway with modifications
  - Replace hostname in media URLs
  - Replace query parameters in media URLs

Clones can appear in Pathway priority list

```
{
        "TITLE": "Content Steering Manifest Example"
    "VERSION": 1,
    "PATHWAY-PRIORITY": [ "PATH-C", "PATH-B", "PATH-A" ],
    "TTL": 20,
    "RELOAD-URI": "./steering-manifest.json?s_weight=10",
    "PATHWAY-CLONES": [
        {
            "ID": "PATH-C",
            "BASE-ID": "PATH-A",
            "URI-REPLACEMENT": {
                "HOST": "pathway-c.example.com",
            }
        }
    ]
}
```

Adopting Content Steering

Host transmission protocol defines binding

- HLS, MPEG-DASH

Content Steering terms are bound to host protocol

- Content Delivery Protocol -> HLS
- Content Description -> HLS Multivariant Playlist
- Pathway ID -> PATHWAY attribute

Outcomes

Offered in iOS 16 (2022) and onwards

+ HLS.js, DASH.js, Video.js, Shaka Player (ExoPlayer in progress)

Vendor support

- Synamedia, Brightcove, System73, Velocix, Bitmovin, ...

Used by major streaming providers

Has remained stable for three years

The spec: draft-pantos-content-steering

Originally specified as part of HLS spec

- draft-pantos-hls-rfc8216bis

Pulled out and abstracted

- No HLS-specific details (or even media streaming)
- Bindings and rules that host spec must implement

Much helpful review & feedback from IETF hls-interest

What's next

Plan to publish as an RFC

Reference for HLS bis RFC, and for MPEG-DASH

HLS published as Independent Submission on Informational track

- Content Steering likewise

Find me later for Q&A