

Informational
Internet-Draft
Intended status: Informational
Expires: 18 September 2026

R. Pantos, Ed.
E. Vershen
Apple Inc.
17 March 2026

Content Steering
draft-pantos-content-steering-03

Abstract

This document describes a mechanism for servers to communicate their preferences to clients about utilizing alternate servers for streaming content. These alternate servers are typically distinct Content Delivery Networks or any other servers that offer similar distribution services. The mechanism described in this document is designed to be universally applicable and independent of any specific Content Delivery Protocol.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 18 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

This Informational Internet-Draft is submitted as an RFC Editor Contribution and/or non-IETF Document (not as a Contribution, IETF Contribution, nor IETF Document) in accordance with BCP 78 and BCP 79.

Table of Contents

1. Introduction	2
2. Definitions and Abbreviations	3
3. CDP Responsibilities	4
4. Steering Manifest	5
5. Pathway Cloning	6
6. Steering Query Parameters	8
7. Steering Client Responsibilities	9
8. Content Steering Manifest Examples	10
8.1. Content Steering Manifest	10
8.2. Content Steering Manifest with Pathway Clone	11
9. Acknowledgements	11
10. Contributors	11
11. IANA Considerations	11
11.1. vnd.apple.steering-list	11
12. Security Considerations	13
13. References	13
13.1. Normative References	13
13.2. Informative References	14
Authors' Addresses	14

1. Introduction

Content Steering provides a simple mechanism for content delivery systems to apportion Content Servers among clients. The Steering Server can divide clients for simple A/B testing, for load balancing, for geographic diversity, or any other criteria. Content Steering is an extension of a Content Delivery Protocol (CDP) and cannot be implemented separately from a specific CDP.

The purpose of this document is to facilitate the use of Content Steering by a variety of Content Delivery Protocols. While this document describes the basic structure of Content Steering, each Content Delivery Protocol using Content Steering is required to elaborate some aspects in their own specifications.

Data SHOULD be carried over HTTP [RFC9112], but, in general, a URI can specify any protocol that can reliably transfer the specified resource on demand. In the case of HTTP URIs the request SHOULD be a GET request.

This informational specification is not an Internet standard. It is discussed on the IETF hls-interest list (<https://mailarchive.ietf.org/arch/browse/hls-interest/>) but does not represent the consensus of the IETF. It was developed by Apple Inc.

2. Definitions and Abbreviations

- * Base Pathway: a Pathway used as the basis for a Pathway Clone.
- * Content Delivery Network (CDN): a geographically distributed network of Content Servers.
- * Content Delivery Protocol (CDP): an application-layer protocol that specifies data structures and access methods for iterative delivery of sequential content.
- * Content Description: a document (or group of documents) that the CDP uses to specify the URIs used to obtain a particular set of content from a server.
- * Content Server: an endpoint that manages the delivery of content by responding to client requests.
- * Content Steering: is a mechanism that enables servers to communicate their preferences to clients regarding the use of alternate Content Servers for streaming content, optimizing delivery based on factors such as network conditions and server load.
- * Pathway: a predetermined route for content delivery, typically through a particular Content Server.
- * Pathway Clone: A Pathway produced by applying Pathway Cloning to a Base Pathway.
- * Pathway Cloning: a process that takes an existing Pathway and a set of modifiers and generates a new Pathway Clone that is not explicitly defined in the original Content Description.
- * Pathway ID: an identifier for a Pathway. A Pathway ID is a non-empty string containing characters from the set [a-z], [A-Z], [0-9], '.', '-', and '_'.

- * Steering Manifest: a JSON document that serves as a dynamic guide for the client. It contains steering instructions and identifies the available Pathways and their priority order.
- * Steering Manifest URI: a unique URI that identifies a specific Steering Manifest on a Steering Server.
- * Steering Server: an endpoint that returns Steering Manifests.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. CDP Responsibilities

A Content Delivery Protocol (CDP) that supports Content Steering MUST:

Associate a distinct Pathway ID with each Pathway.

Define which content is subject to Content Steering and how to associate a particular content URI with a particular Pathway ID.

Declare an initial Steering Manifest URI in each Content Description.

If the CDP allows Pathway Cloning, define how to incorporate Pathway Clones into the Content Description.

A CDP that allows Content Steering MAY also specify:

An initial active Pathway in each Content Description.

CDP specific extensions to URI-REPLACEMENT object within the Pathway Clone object. See Pathway Cloning (Section 5).

CDP specific query parameters. See Steering Query Parameters (Section 6).

Whether clients must support Content Steering is a design decision left to the CDP implementation.

Two widely-used CDPs that implement Content Steering are: HTTP Live Streaming (HLS [RFC8216bis]) and Dynamic Adaptive Streaming over HTTP (DASH [ISO_23009_1]).

4. Steering Manifest

Content Steering is accomplished by having clients periodically read a Steering Manifest from a Steering Server. The initial Steering Manifest URI is determined by the Content Description. The initial Steering Manifest reload interval is determined by the CDP. Subsequent URIs and reload intervals are determined by the most recently fetched Steering Manifest. The Steering Manifest identifies the available Pathways and their priority order.

The Steering Manifest response is a JSON [RFC8259] object. In keeping with the JSON standard the text MUST be encoded using UTF-8 [RFC3629]. The JSON object has the form:

```
{
  "VERSION": number,
  "TTL": number,
  "RELOAD-URI": string,
  "PATHWAY-PRIORITY":
  [
    One or more Pathway IDs in order of preference
  ],
  "PATHWAY-CLONES":
  [
    One or more Pathway Clone objects.
  ]
}
```

The JSON object MAY contain other key:value pairs. In particular, a CDP MAY define additional keys, including in any contained objects. Keys defined by a CDP MUST be defined with reference to a specific Steering Manifest VERSION number defined by this document. A client MUST ignore any key of the Steering Manifest that it does not recognize. Note that keys are case-sensitive.

***VERSION*:** An integer that specifies the version of Steering Manifest. This specification defines Steering Manifest VERSION 1. A client MUST refuse to use Steering Manifest if the VERSION is missing or the VERSION number is unrecognized. This element is REQUIRED.

***TTL*:** An integer that specifies the number of seconds the client MUST wait after loading the Steering Manifest before reloading it. The recommended value is 300 seconds (5 minutes). The Steering Server MAY vary the TTL per client to distribute server load. This element is REQUIRED.

RELOAD-URI: A string that specifies the URI the client MUST use for future Steering Manifest requests. The RELOAD-URI MAY be relative to the current Steering Manifest URI. Certain URI schemes (such as "data") cannot act as base URIs for relative URIs. Attempting to specify a relative URI in that case MUST produce an error. This element is OPTIONAL.

PATHWAY-PRIORITY: An array of string elements, where each string represents a Pathway ID. Elements in the PATHWAY-PRIORITY array are ordered by Pathway preference, with the first being most preferred. A Steering Manifest MUST contain at least one Pathway. A Pathway ID in the PATHWAY-PRIORITY array MUST NOT appear more than once. A Pathway is recognized only if it is specified by the Content Description or in the current Steering Manifest. Clients MUST ignore unrecognized Pathway IDs in the PATHWAY-PRIORITY array. This element is REQUIRED.

Note that it is important for the most-preferred Pathway of the initial Steering Manifest to agree with the initial Pathway in the Content Description. Immediately redirecting a client to a different Pathway on startup will delay content delivery and increase network utilization.

PATHWAY-CLONES: An array of Pathway Clone objects. See Pathway Cloning (Section 5). If present, the array must contain at least one element. This element is OPTIONAL.

5. Pathway Cloning

A Steering Server can introduce novel Pathways by cloning existing ones. A Pathway Clone is produced by taking an existing Pathway and applying well-defined replacements to the URIs of every Pathway member.

A Pathway Clone object is a JSON object that has the following form:

```
{
  "BASE-ID": string,
  "ID": string,
  "URI-REPLACEMENT":
  {
    "HOST": string,
    "PARAMS":
    {
      key:value pairs for query parameters
    }
  }
}
```

As part of the Steering Manifest, the Pathway Clone object MAY contain other key:value pairs, see Section 4.

BASE-ID: A string that specifies the Pathway ID of the Base Pathway. This element is REQUIRED.

ID: A string that specifies the Pathway ID for the Pathway Clone. The ID specified for a Pathway Clone MUST NOT match any other Pathway ID. This element is REQUIRED.

URI-REPLACEMENT: An object that defines URI modifications to apply during the cloning process. This element is REQUIRED.

HOST: A string that specifies the Hostname for cloned URIs. This element is OPTIONAL.

PARAMS: A JSON object that specifies query parameters for cloned URIs. The keys represent query parameter names, and the values correspond to the associated parameter values. This element is OPTIONAL.

Clone a Pathway by following these steps:

1. Identify the Base Pathway by matching its ID to the value of the BASE-ID key in the Pathway Clone object.
2. Duplicate the Base Pathway.
3. Set the ID of the new Pathway to the value of the ID key in the Pathway Clone object.
4. For each URI belonging to the new Pathway:
 - A. Resolve the URI against its base if necessary and then replace its hostname with the URI-REPLACEMENT HOST string (if present).
 - B. Append each name/value pair of the PARAMS object (if present), in the UTF-8 order of the names, as a query parameter of the URI. If a parameter of the same name is already present in the URI, replace it with the one from the PARAMS object.
 - C. Replace the URI in the new Pathway with the modified URI.

A CDP that extends the URI-REPLACEMENT element MUST modify the above process to include its extensions.

A Pathway Clone MAY use another Pathway Clone as its base if it appears earlier in the PATHWAY-CLONES array.

The Pathway ID that is defined by a Pathway Clone MAY appear in the PATHWAY-PRIORITY list, in any position.

The HOST string of a Pathway Clone MUST be non-empty if it is present.

The name of a PARAMS object name/value pair MUST be non-empty. The names and values in a PARAMS object MUST be able to form a valid URI query parameter. Any reserved characters in those strings MUST be percent-encoded [RFC3986].

A client that does not have the definition of a Pathway specified by the BASE-ID string of a Pathway Clone object MUST ignore the Pathway Clone. The client has the definition of a Pathway if it appears in the original Content Description, or appears in the Pathway Clones array from the current Steering Manifest.

Client support of Pathway Cloning is OPTIONAL, unless determined otherwise by the CDP. A steering server SHOULD ensure that the PATHWAY-PRIORITY list always contains at least one Pathway defined in the original Content Description.

6. Steering Query Parameters

The client sends a request with the Steering Manifest URI to obtain the Steering Manifest. Each CDP can define query parameters that the client SHOULD add to the URI.

To support stateless processing by the Steering Server the following client state SHOULD be provided:

The ID of the Pathway currently applied by the client.

A current prediction of content download throughput made by the client for the currently applied Pathway. The exact method of bit rate estimation will vary by client, but the units (such as bits per second) SHOULD be specified by the CDP.

HTTP proxy caches SHOULD be configured to exclude highly variable query parameters from their cache keys, or treat the Steering Manifest response as non-cacheable.

7. Steering Client Responsibilities

A Pathway is applied by first choosing a particular Pathway ID. The set of URIs to which the client is allowed to use is then restricted to those belonging to that Pathway. If a client is currently using a content URI that does not belong to the applied Pathway, it **MUST** switch to using one that does.

A client that supports Content Steering **MUST** implement the following algorithm. In this algorithm the exact meaning of "being penalized" is specific to the particular CDP, but its general meaning is that the Pathway will not be used by the client for some period. A CDP **SHOULD** add clarifications and/or changes to this algorithm as is appropriate for its Content Delivery Protocol:

1. When using a Content Description that contains an initial Steering Manifest URI, load the Steering Manifest. If permitted by the CDP, a client that wishes to use the content before it obtains the Steering Manifest **SHOULD** apply the initial Pathway of the Content Description. If the Content Description does not contain a initial Pathway, the client **MAY** use any Pathway until it obtains the Steering Manifest.
2. When a Steering Manifest is received, perform Pathway Cloning on any members of the PATHWAY-CLONES array. Then, perform a Content Steering evaluation (step 5).
3. If all the URIs from the current Pathway fail with a network error, or if the lowest-bitrate URI or combination of URIs cannot be downloaded quickly enough to support real-time use, the client **MAY** mark the current Pathway as penalized, and perform a Content Steering evaluation (step 5).
4. If a Pathway has been penalized long enough that it may have recovered, the client **SHOULD** un-penalize that Pathway and perform a Content Steering evaluation (step 5). The algorithm the client uses for determining the penalization duration **MAY** be defined by the CDP. If not, it is determined by the client.
5. Content Steering evaluation: If no Pathway is currently applied, or the current Pathway is not the first in the PATHWAY-PRIORITY list, or is no longer on the list, or is being penalized, then apply the first non-penalized Pathway on the list. If no such Pathway is available, the client **SHOULD** remain on the current Pathway.

6. When the current Steering Manifest expires, as defined by the TTL attribute, issue a new Steering Manifest request for the URI specified by RELOAD-URI or the previous server URI if none. The RELOAD-URI may be absolute or relative to the previous server URI.
7. When the Steering Manifest cannot be loaded or parsed correctly:
 - A. If the client receives HTTP 410 Gone in response to a manifest request, it MUST NOT issue another request for that URI for the remainder of the session. It MAY continue to use the most-recently obtained list of Pathway priorities.

If no list of Pathways priorities is available (for example, a 410 on the initial Manifest request), the client SHOULD create a priority list from the Pathways in the Content Description, with the initial Pathway, if any, as the highest priority. The client should proceed with the portions of this algorithm that do not involve requesting Steering Manifests.
 - B. If the client receives HTTP 429 Too Many Requests with a Retry-After header in response to a manifest request, it SHOULD wait until the time specified by the Retry-After header to reissue the request.
 - C. Otherwise, the client SHOULD continue to use the previous values and attempt to reload it after waiting for the previously-specified TTL. If no Steering Manifest has been successfully parsed, a TTL of 5 minutes SHOULD be used.

8. Content Steering Manifest Examples

8.1. Content Steering Manifest

This example shows a Content Steering Manifest.

```
{
  "VERSION": 1,
  "TTL": 300,
  "RELOAD-URI": "https://example.com/steering?video=00012&session=123",
  "PATHWAY-PRIORITY": [
    "CDN-A",
    "CDN-B"
  ]
}
```

8.2. Content Steering Manifest with Pathway Clone

This example extends the previous example with a Steering Manifest that includes a Pathway Clone.

```
{
  "VERSION": 1,
  "TTL": 300,
  "PATHWAY-PRIORITY": [
    "CDN-A-CLONE",
    "CDN-A"
  ],
  "PATHWAY-CLONES": [
    {
      "BASE-ID": "CDN-A",
      "ID": "CDN-A-CLONE",
      "URI-REPLACEMENT": {
        "HOST": "backup2.example.com",
        "PARAMS": {
          "token": "dkfs1239414"
        }
      }
    }
  ]
}
```

9. Acknowledgements

Thanks to the members of the IETF hls-interest mailing list for feedback.

10. Contributors

Significant contributions to the design of this protocol were made by Naiwei Zheng and Jordan Schneider.

Special thanks to Will Law, Jason Thilbeault, Yuriy Reznik, Phil Cluff, Pieter-Jan Speelmans, Allan Zhou, and other members of the hls-interest group

11. IANA Considerations

11.1. vnd.apple.steering-list

IANA has registered the following media type [RFC2046]:

Type name: application

Subtype name: vnd.apple.steering-list

Required parameters: none

Optional parameters: none

Encoding considerations: encoded as JSON (UTF-8), which is 8-bit text. This media type may require encoding on transports not capable of handling 8-bit text. See Section 4 for more information.

Security considerations: See Section 12.

Compression: this media type does not employ compression.

Interoperability considerations: There are no byte-ordering issues since files are 8-bit text. Applications could encounter unrecognized keys, which SHOULD be ignored.

Published specification: see Section 4.

Applications that use this media type: streaming media content delivery protocols such as HLS and DASH, including media servers and media players.

Fragment identifier considerations: no Fragment Identifiers are defined for this media type.

Query parameter considerations: this specification does not define any query parameters, however any CDP using Content Steering may define query parameters. (See Section 6.)

Additional information:

Deprecated alias names for this type: none

Magic number(s): none

File extension(s): .json (see Section 4)

Macintosh file type code(s): none

Person & email address to contact for further information: Dimitri Podborski, dpodborski AT apple.com.

Intended usage: LIMITED USE

Restrictions on usage: none

Author: Roger Pantos

Change Controller: Dimitri Podborski

12. Security Considerations

Since the protocol generally uses HTTP to transfer data, most of the same security considerations apply. See Section 15 of HTTP [RFC9112].

Parsers are typically subject to "fuzzing" attacks. Implementors SHOULD pay particular attention to code that will parse data received from a server and ensure that all possible inputs are handled correctly.

Content Description files contain URIs, which clients will use to make network requests of arbitrary entities. Clients SHOULD range-check responses to prevent buffer overflows. See also the Security Considerations section of "Uniform Resource Identifier (URI): Generic Syntax" [RFC3986].

Apart from URI resolution, this format does not employ any form of active content.

HTTP requests often include session state ("cookies"), which may contain private user data. Implementations MUST follow cookie restriction and expiry rules specified by "HTTP State Management Mechanism" [RFC6265] to protect themselves from attack. See also the Security Considerations section of that document, and "Use of HTTP State Management" [RFC2964].

13. References

13.1. Normative References

- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/info/rfc2046>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2964] Moore, K. and N. Freed, "Use of HTTP State Management", BCP 44, RFC 2964, DOI 10.17487/RFC2964, October 2000, <<https://www.rfc-editor.org/info/rfc2964>>.

- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <<https://www.rfc-editor.org/info/rfc3629>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, DOI 10.17487/RFC6265, April 2011, <<https://www.rfc-editor.org/info/rfc6265>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <<https://www.rfc-editor.org/info/rfc9112>>.

13.2. Informative References

- [ISO_23009_1]
International Organization for Standardization,
"Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats", ISO/IEC International Standard 23009-1:2022, August 2022, <<https://www.iso.org/standard/83314.html>>.
- [RFC8216bis]
Pantos, R., Ed., "HTTP Live Streaming 2nd Edition", February 2026, <<https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis-20>>.

Authors' Addresses

Roger Pantos (editor)
Apple Inc.
Cupertino, California
United States
Email: http-live-streaming-review@group.apple.com

Eryk Vershen
Apple Inc.
Cupertino, California
United States
Email: content-steering-review@group.apple.com