New Candidate Technical Specification

DASH-IF CTS Part XX rev 1 Current version: 0.9.6

Status:		Draft		Internal Review	X	Community Review		Editor's Proposal		Agreed	

Title:	Content Steering for DASH		
Source:	DASH-IF Interoperability Working Group		
Supporting Companies:	Akamai, Disney Streaming, Comcast, AWS Elemental, Qualcomm		
Category:	**Candidate Technical Specification**	***Date:***	2022-12-21
Abstract:	Content distributors often use multiple Content Delivery Networks (CDNs) to distribute their content to the end-users. They may upload a copy of their catalogue to each CDN, or more commonly have all CDNs pull the content from a common origin. Alternate URLs are generated, one for each CDN, that point at identical content. DASH players may access alternate URLs in the event of delivery problems. Content steering describes a deterministic capability for a content distributor to switch the content source that a player uses either at start-up or midstream, by means of a remote steering service. The DASH implementation of Content Steering also supports the notion of a proxy steering server which can switch a mobile client between broadcast and unicast sources.		
Disclaimer:	This document is a candidate Technical Specification. DASH-IF is expecting to publish this initially, but to submit the specification to a formal specification organization. The primary choice is ETSI, for which DASH-IF has a PAS agreement. This document is not yet final. It is provided for public review until the deadline mentioned below. If you have comments on the document, please submit comments by one of the following means: - at the github repository https://github.com/Dash-Industry-Forum/Content-Steering/issues, or - the mailing list at iop@dashif.org Please add a detailed description of the problem and the comment. Based on the received comments a final document will be published latest by the expected publication date below if the following additional criteria are fulfilled: - All comments from community review are addressed - A time plan for test, conformance and reference tools are available. This includes availability of test services and an implementation oin the dash.js reference tools		
Commenting Deadline:	Feb 15, 2023		
Expected Publication:	Mar 31, 2023		
Other Comments	Beyond this specification, it is expected that DASH-IF IOP Guidelines are updated to reference this specification, in particular the client requirements.		

DASH-IF CTS 00XX V0.9.6 (2022-12)

DASH-IF Candidate Technical Specification:
Content Steering for DASH

DASH Industry Forum

3855 SW 153rd Dr.

Beaverton, OR 97003 - USA

Email : admin@dashif.org

Important notice

The present document can be downloaded from:
http://www.dashif.org/guidelines

Contents

Intellectual Property Rights

Disclaimer

This is a document made available by DASH-IF. The technology embodied in this document may involve the use of intellectual property rights, including patents and patent applications owned or controlled by any of the authors or developers of this document. No patent license, either implied or express, is granted to you by this document. DASH-IF has made no search or investigation for such rights and DASH-IF disclaims any duty to do so. The rights and obligations which apply to DASH-IF documents, as such rights and obligations are set forth and defined in the DASH-IF Bylaws and IPR Policy including, but not limited to, patent and other intellectual property license rights and obligations. A copy of the DASH-IF Bylaws and IPR Policy can be obtained at http://dashif.org/.

The material contained herein is provided on an "AS IS" basis and to the maximum extent pe mitted by applicable law, this material is provided AS IS, and the authors and developers of this material and DASH-IF hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of workmanlike effort, and of lack of negligence.

In addition, this document may include references to documents and/or technologies controlled by third parties. Those third-party documents and technologies may be subject to third party rules and licensing terms. No intellectual property license, either implied or express, to any third-party material is granted to you by this document or DASH-IF. DASH-IF makes no warranty whatsoever for such third-party material.

Note that technologies included in this document and for which no test and conformance material is provided, are only published as candidate technologies, and may be removed if no test material is provided before releasing a new version of this guidelines document. For the availability of test material, please check https://www.dashif.org.

Foreword

This Technical Specification (TS) has been produced by the DASH-IF Technical Working Group.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in deliverables except when used in direct citation.

Executive summary

Content distributors often use multiple Content Delivery Networks (CDNs) to distribute their content to the end-users. They may upload a copy of their catalogue to each CDN, or more commonly have all CDNs pull the content from a common origin. Alternate URLs are generated, one for each CDN, that point at identical content. DASH players may access alternate URLs in the event of delivery problems. Content steering describes a deterministic capability for a content distributor to switch the content source that a player uses either at start-up or midstream, by means of a remote steering service. The DASH implementation of Content Steering also supports the notion of a proxy steering server which can switch a mobile client between broadcast and unicast sources.

1 Scope

The present document specifies Content Steering for DASH.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, DASH-IF cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document:

[1] HTTP Live Streaming 2nd Edition, https://datatracker.ietf.org/doc/html/draft-pantos-hls-rfc8216bis-10

[2] ISO/IEC 23009-1:2021: "Information technology -- Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats"

[3] RFC3986 Uniform Resource Identifier (URI): Generic Syntax https://datatracker.ietf.org/doc/html/rfc3986

[4] RFC9110 HTTP Semantics https://www.rfc-editor.org/rfc/rfc9110.html

[5] RFC6585 Additional HTTP Status Codes https://www.rfc-editor.org/rfc/rfc6585

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

[i.1] DASH-IF Interoperability Points, Part 1: Overview, architecture and interfaces

[i.2] ISO/IEC 14496-12:2021 "Information technology -- Coding of audio-visual objects -- Part 12: ISO base media file format"

[i.3] ISO/IEC 23000-19:2020 "Information technology -- Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media"

[i.4] CTA-5004: Web Application Video Ecosystem-Common Media Client Data. [Online] Available: https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5004-final.pdf

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

Content Steering Server: A network element that provides steering information to one or several or many DASH Players for DASH operation across multiple CDNs.

DASH Content Steering Manifest: A document that includes steering instructions to a DASH player provided by a Content Steering Server

3.2 Symbols

For the purposes of the present document, the following symbols apply:

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CDN	Content Delivery Network
CMAF	Common Media Application Format
CMCD	Common Media Client Data
CTA	Consumer Technology Association
DASH	Dynamic Adaptive Streaming over HTTP
DCSM	DASH Content Steering Manifest
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ISO	International Standards Organization
MPD	Media Presentation Description
TTL	Time-To-Live
URL	Uniform Resource Locator

4 Overview and Architecture

Content distributors often use multiple Content Delivery Networks (CDNs) to distribute their content to the end-users as shown in Figure 4-1. They may upload a copy of their catalogue to each CDN, or more commonly have all CDNs pull the content from a common origin. In the DASH Media Presentation Description (MPD) as defined in ISO/IEC 23009-1 [2], multiple URLs are provided, one for each CDN, that point at identical content. Typically, a DASH player will access content from one single location, using the default location defined by the MPD.

If the DASH player then observes delivery problems, it may chose to access content via the alternate URLs. This operation is completely client-driven, is not standardized between players and may not be the desired behaviour of the content distributor.

Content steering provides a deterministic capability for a content distributor to switch the content source that a player uses either at start-up or midstream, by means of a remote steering service.

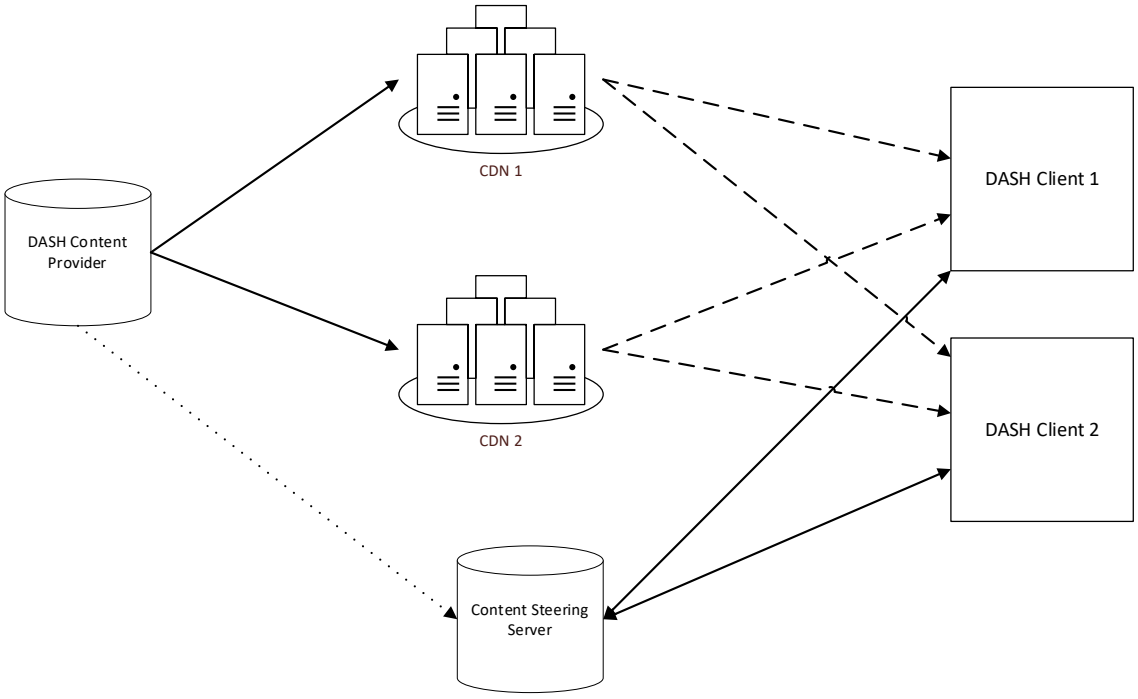

Figure 4-1 Basic Architecture with Content Steering

Steering is accomplished by having the DASH client periodically access a content steering server to retrieve a steering manifest, which instructs the player as to the availability and priority of content sources.

The typical procedures followed when content steering is in use are shown in Figure 4-2 for the case when the content is provided on two CDNs. The DASH content provider generates an MPD that includes Base URLs to CDN1 and CDN2, as well as an address where the clients can access the content steering server. The provider also uploads the MPD and the Content segments to both CDNs. At the start of playback, the DASH client requests the MPD from one CDN, in this case from CDN2. It finds the content steering server URL, and it may find information that instructs it to contact the content steering server prior to the first segment request versus the default behaviour of making the request once its starting buffer is full. The player then makes a request to the content steering server. The content server responds with a content steering manifest and the DASH client uses the information within to prioritize the segment source, in this case from CDN2. After some time, the content provider may collect operational information from the participating clients, for example by using Common Media Client Data (CMCD) as defined in CTA-5004 [i.4]. Based on this information, the content provider may update the content steering server, and based on this updated information, the content steering manifest may change. When the client requests an update to the content steering manifest, new information may be provided that instructs the DASH client to request the Segments from CDN1 instead of CDN2. The DASH client then switches smoothly, at a segment boundary, to download the Segments from CDN1 instead of CDN2. The steering server response can also be used to steer the DASH client between alternate sources for DASH manifest refreshes, via service descriptors contained within the **`Location`** element.

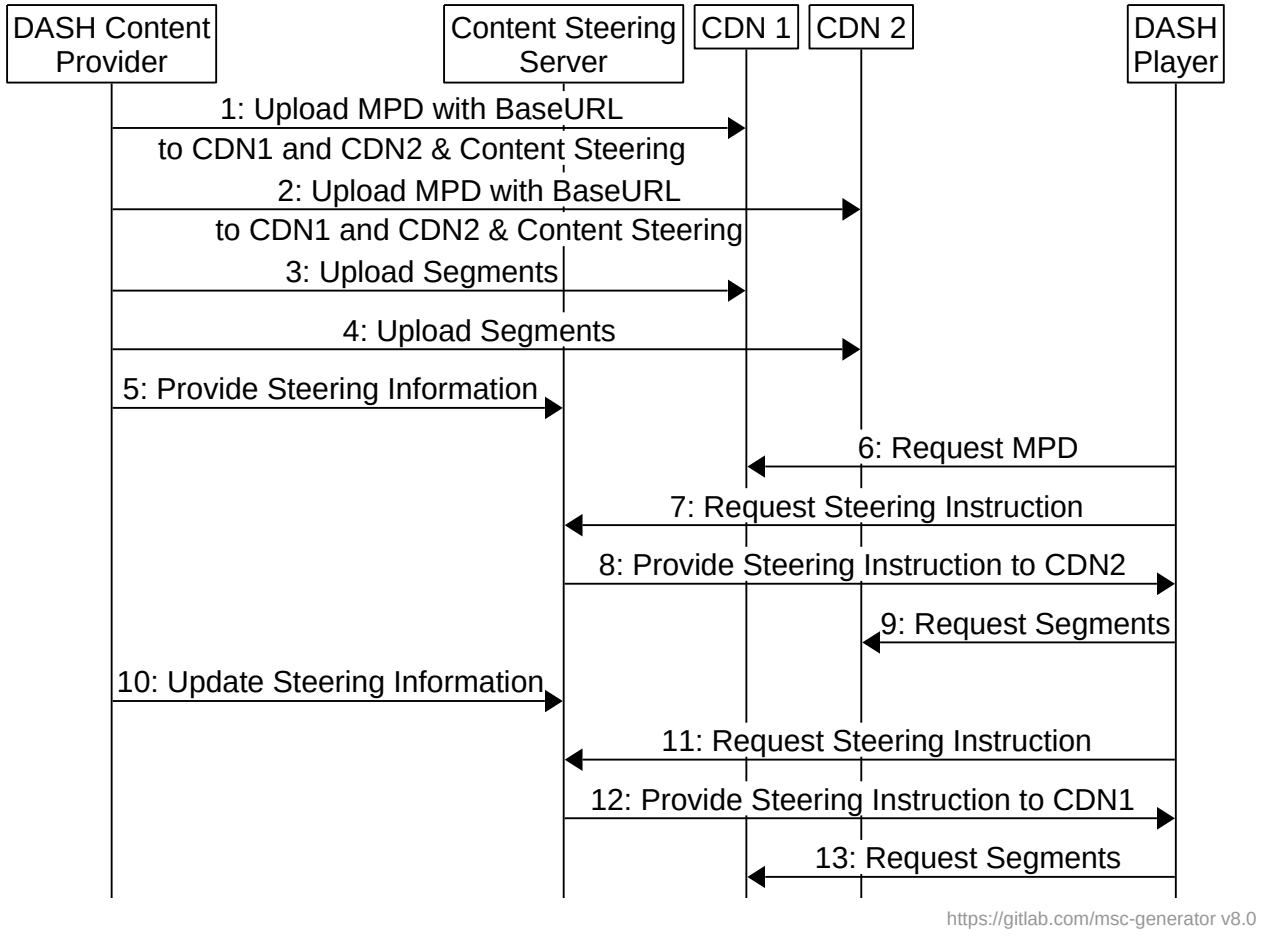

Figure 4-2 Typical procedures in Content Steering operation

This specification provides the following detailed information to support interoperable operation of content steering in DASH Media Presentations:

- Clause 5 provides the details on how to signal different alternate CDNs in the DASH MPD using existing **`BaseURL`** elements as well as an extension to the DASH MPD in order to provide the location of the Content Steering server as well as some additional instructions to the client. This includes the semantics and XML syntax of the new **`ContentSteering`** element, as well as the extension of the **`Location`** element to add a `@serviceLocation` attribute.

- Clause 6 provides the details on the DASH content steering manifest (DCSM) and the behaviour of the content steering server, in particular the semantics and JSON syntax of the steering server response.
- Clause 7 defines the DASH player behaviour in detail. A DASH player supporting Content Steering is required to implement the detailed procedures documented in this clause.
- Clause 8 provides guidelines on the usage of DASH Annex I for providing the client identity as part of the DASH player to content steering server communication.
- Annex A provides operational examples of a DASH MPD as well as of the response of the content steering server.

5 Content Steering Signalling in DASH MPD

5.1 Overview

Content distributors often use multiple Content Delivery Networks (CDNs) to distribute their content to the end-users. They may upload a copy of their catalogue to each CDN, or more commonly have all CDNs pull the content from a common origin. Alternate URLs are generated, one for each CDN, that point at identical content.

The DASH MPD supports the **`BaseURL`** element to allow the listing of these alternate URLs, as well as the **`Location`** element for pointing at alternate manifest sources. DASH players may access alternate URLs in the event of delivery problems. Content steering describes a deterministic capability for a content distributor to switch the content source that a player uses either at start-up or midstream, by means of a remote steering service.

> NOTE: Overlapping `@serviceLocation` names for steerable and non-steerable BaseURLs is discouraged. i.e. if a server-side ad insertion provides BaseURLs with an overlapping `@serviceLocation` value it could cause unintended behavior.

To enable content steering, a new **`ContentSteering`** element is introduced in the MPD, the **`Location`** element is updated to include a new `@serviceLocation` attribute and Annex I is extended to allow the flexible insertion of URL parameters into the new **`ContentSteering`** element.

> NOTE: These changes impact the MPEG DASH MPD schema. DASH-IF is aware that such an extension needs to be coordinated with MPEG to address this in ISO/IEC 23009-1. MPEG is asked to comment on this proposal and potentially update the MPEG DASH specification to support the functionality. Based on the feedback from MPEG, the implementation of the functionality into the schema may slightly change.

The DASH MPD schema is expected to be extended as follows:

```
<xs:complexType name="MPDtype">
…
<xs:sequence>
    …
    <xs:element name="BaseURL" type="BaseURLType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Location" type="LocationType" minOccurs="0" maxOccurs="unbounded"/>
    …
    <xs:element name="UTCTiming" type="DescriptorType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="ContentSteering" type="ContentSteeringType" minOccurs="0" maxOccurs="1"/>
    …
</xs:sequence>

<!-- Base URL -->
  <xs:complexType name="BaseURLType">
    <xs:simpleContent>
      <xs:extension base="xs:anyURI">
        <xs:attribute ref="serviceLocation"/>
       …
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

<xs:attribute name="serviceLocation">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9_.-]"/>
      <xs:maxLength value="64" />
    </xs:restriction>
```

```
  </xs:simpleType>
</xs:attribute>

<!-- LocationType -->
<xs:complexType name="LocationType">
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:attribute ref="serviceLocation"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<!--ContentSteering -->
<xs:complexType name="ContentSteeringType">
    <xs:simpleContent>
        <xs:extension base="xs:anyURI">
            <xs:attribute ref="defaultServiceLocation"/>
            <xs:attribute name="queryBeforeStart" type="xs:boolean" default="false"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:attribute name="defaultServiceLocation">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9_.-,]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

The **ContentSteering** element defines the URL of a steering server. Additional optional attributes may be provided in the Content Steering element in order to define a default service location and to provide an indication as to whether the content steering server needs to be contacted prior to requesting the first Segment. For details refer to clause 5.2.

The **ContentSteering** element shall appear at most once in an MPD and only at the MPD level. The **ContentSteering** element shall be the last sequenced element within the MPD level.

The semantics of the attributes and elements for Content Steering are provided in subclause 5.2, Table 5.2-1 and for the **Location** element in subclause 5.2, Table 5.2-2.

5.2 Semantics

Table 5.2-1 — Semantics of the Content Steering element

Element or Attribute Name	Use	Description
ContentSteering		A URL that can be used to access the Content Steering server. The URL points to a DASH Content Steering Manifest (DCSM) as defined in clause 6.
@defaultServiceLocation	O	This attribute specifies a comma separated list of **BaseURL**@serviceLocation and **Location**@serviceLocation IDs that the client can use to select Location and BaseURL elements with which to start playback.
@queryBeforeStart	OD Default: false	If true, specifies that the player must resolve the response from the Steering Server prior to starting playback. Default value is false.
Key For attributes: M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory For elements: <minOccurs>...<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an @.		

Table 5.2-2 — Semantics of the Location element

Element or Attribute Name	Use	Description
Location		A URL that can be used to access the Content Steering server. The URL points to a DASH Content Steering Manifest (DCSM) as defined in clause 6.
@serviceLocation	O	This attribute specifies a relationship between Location elements such that Location elements with the same @serviceLocation value are likely to have their URLs resolve to services at a common network location, for example a common Content Delivery Network. Within a manifest, Location@serviceLocation attributes must be unique against any BaseURL@serviceLocation attributes. For interoperability with other formats, the service location string shall only contain characters from the set [a..z], [A..Z], [0..9], '.', '-', and '_'.

6 DASH Steering Manifest and Server Behaviour

6.1 Overview

Based on the description in clause 4, the DASH Steering server provides a Steering Manifest on a request from a DASH client. The DASH Content Steering Manifest (DCSM) is a json document and shall be formatted according to the JSON schema provided in clause 6.2.

The semantics of the key-value pairs of the DCSM are defined in clause 6.3.

A client shall ignore any key of the DCSM that it does not recognize. DCSM keys are case-sensitive.

Note: this structure is intentionally similar to that defined by [1] Section 7.1. for HLS for the purposes of interoperability. The DASH variant and versioning are defined in this document and intentionally exclude features in the HLS design (such as pathway cloning) which are not applicable to DASH.

6.2 JSON Syntax

```
{
   "VERSION": number,                    // REQUIRED, must be an integer
   "TTL": number,                        // REQUIRED, number of seconds
   "RELOAD-URI": string,                 // OPTIONAL, URI
   "PATHWAY-PRIORITY": [       // REQUIRED, array of serviceLocation
identifiers in order of preference ]
}
```

6.3 Semantics

The semantics of the DCSM are defined in Table 6.3-1.

Table 6.3-1 — Semantics of the DASH Content Steering Manifest

Key	Use	Description
DCSM		A URL that can be used to access the Content Steering server. The URL points to a DASH Content Steering Manifest (DCSM) as defined in clause 6.

Key			Use	Description
		VERSION	M	The version of DCSM. This specification defines DASH Steering Manifest version 1. A client shall refuse to use a DCSM with an unrecognized value for this key.
		TTL	M	Specifies how many seconds the client shall wait before reloading the DCSM. The recommended value is 300 seconds . The Steering Server may vary the TTL by client and the TTL may vary with each reload of the steering server manifest
		RELOAD-URI	O	If present, specifies the URI the client shall use the next time it obtains the DCSM. The RELOAD-URI may be relative to the current DCSM Manifest URI. If not present, the current DCSM Manifest URI shall be used.
		PATHWAY-PRIORITY	OD Default [] PATHWAY-PRIORITY is an array of **BaseURL**@serviceLocation and **Location**@serviceLocation IDs. Elements in the PATHWAY-PRIORITY array are ordered by preference of the serviceLocation to be selected, with the first being most preferred. If present, the value of this key shall contain at least one element for serviceLocation. A @serviceLocation ID in the PATHWAY-PRIORITY array shall not appear more than once. Clients shall ignore unrecognized @serviceLocation IDs in the PATHWAY-PRIORITY array.	
Key M=mandatory, O=optional, OD=optional with default value, CM=conditionally mandatory				

7 Normative DASH Client Steering behaviour

A DASH client supporting DASH Content Steering as defined in this specification shall adhere to the following procedures:

1. If the **ContentSteering** element is present in the MPD, then the client shall parse the element and extract the server URI, as well as the @defaultServiceLocation, and @queryBeforeStart attributes, if present.
2. If any extended HTTP GET request parametrization instructions as defined in clause 8 are present in the MPD which target the **ContentSteering** element, then they shall be executed at this stage to modify the server URI.
3. The client shall use the server URI as the STEERING-SERVER-URL.

4. If `@queryBeforeStart` is absent, or present and set to `false`, then the client shall follow its default start-up sequence. Once playback has started and the client has reached its target buffer, it shall proceed to the next step.
5. The client shall make a `GET` request to the `STEERING-SERVER-URL`.
6. The `GET` request should be accompanied by two optional query parameters:
    a. The `_DASH_pathway` parameter shall contain a value of the currently selected **`BaseURL`**`@serviceLocation`, contained in double-quotes. If playback has not yet started due to this being the first request with `@queryBeforeStart` set to `true`, then the `_DASH_pathway` parameter shall be omitted. If multiple `@serviceLocation` attributes are being currently matched, for example in the Location element and in a **`BaseURL`** element, or if multiple `@serviceLocation` attributes have been matched since the last steering server request, then all matched `@serviceLocation` attributes shall be listed, separated by a comma delimiter and contained within a single set of double-quotes.
    b. The `_DASH_throughput` parameter represents a current prediction of media download throughput observed by the client, in units of integer bits per second, from the applied `@serviceLocation`. The exact method of bit rate estimation may vary by client. If the client has multiple throughput estimates available, for example from demuxed audio and video downloads, then it should report the higher of the available estimates. If playback has not yet started due to this being the first request with `@queryBeforeStart` set to true, or if the throughput value is unknown for other reasons, then the `_DASH_throughput` parameter should be omitted. If the `_DASH_pathway` parameter value references multiple serviceLocations, then the `_DASH_throughput` parameter value shall consistent of a comma separated list of throughput values, sequenced such that the n'th item in the `_DASH_throughput` list applies to the n'th item in the `_DASH_pathway` list.
7. Upon receipt of the steering server response, the client should parse it and retrieve the VERSION, TTL, `PATHWAY-PRIORITY` array and optional `RELOAD-URI`. The client shall ignore any steering manifest keys it does not recognize. Manifest keys are case-sensitive.
8. The client sets a timer to re-request the `STEERING-SERVER-URL` after `TTL` seconds.
9. If `RELOAD-URI` is present, then the client shall update the `STEERING-SERVER-URL` to match that specified by `RELOAD-URI`. The `RELOAD-URI` may be relative to the current server URI.
10. If the `VERSION` is a value other than 1, then the client shall abort any further steering behavior.
11. The string entries in the `PATHWAY-PRIORITY` array represent a prioritized list of serviceLocations from which playback should take place, with the highest priority option listed first. This highest priority item is termed the preferred service location. The `PATHWAY-PRIORITY` array is applicable across all periods present in the manifest as long as they include one or more **`BaseURL`** elements with a `@serviceLocation` attribute value included in the `PATHWAY-PRIORITY` array, or inherit from top level **`BaseURL`** elements with a `@serviceLocation` attribute value included in the `PATHWAY-PRIORITY` array.
12. If the client is playing content defined by a **`BaseURL`** element with a `@serviceDescription` not equal to the preferred service location, then the client shall switch at the next segment load to retrieving future content from the **`BaseURL`** referenced by the preferred service location. Note that existing requests against a prior `@serviceLocation` should be allowed to complete and forward buffers should not be trimmed. If the preferred service location is not described in the DASH MPD, then the client should attempt to switch to the next highest priority service location. If no `@serviceLocation` attributes match the manifest, then the client may ignore the current `PATHWAY-PRIORITY` array and make a default decision about which **`BaseURL`** to use. It should still reload the `RELOAD-URI` after the specified `TTL` interval in case new service locations are added.
13. If the manifest contains one or more **`Location`** elements with `@serviceLocation` attributes, then prior to a manifest update, the client shall evaluate the `@serviceLocation` attributes against the PATHWAY-PRIORITY array and select the Location element with the highest matching `@serviceLocation`. If no `@serviceLocation` attributes match then the client may make a default decision about which **`Location`** element to select.
14. A HTTP 410 [4] response from the steering server indicates that access to the steering server is no longer available and that this condition is likely to be permanent. As a result, if the client has a previous valid `PATHWAY-PRIORITY` array from the steering server, it should continue to enforce that prioritization for the

remainder of playback and should cancel any all future reloads . If the `410` response is received on the first steering server request, then the client should abandon all steering behaviors, cancel all future requests and proceed with playback as if the **`ContentSteering`** element were not present.

15. A HTTP `429` response [5] from the steering server indicates that the server has received too many requests. The client should react by substituting the parsed TTL value with the `429 Retry-After` value, if present.
16. If the client encounters playback problems which would normally cause it to try an alternate **`BaseURL`**, it may continue to make that local switching decision, while following these constraints:
    a. The client may only try **`BaseURL`**`@serviceLocation` attributes which were present in the last steering server response.
    b. The client shall try these serviceLocations in the order in which they were prioritized in the last steering server response.
    c. As it switches away from the highest priority `@serviceLocation` for local performance reasons, it shall exclude that `@serviceLocation` for a time-limited period equal to the last steering server TTL that it received. Effectively this means that if the next steering server response again assigns the excluded `@serviceLocation` as the highest priority, the client shall ignore that instruction and instead process the `PATHWAY-PRIORITY` array as if the excluded `@serviceLocation` were not present.
17. If the client encounters playback problems which would normally cause it to try an alternate **`Location`**, it may continue to make that local switching decision, while following these constraints:
    a. The client may only try **`Location`**`@serviceLocation` attributes which were present in the last steering server response. If no `@serviceLocation` attributes match, then the client may make a default decision as to which **`Location`** element to use.
    b. The client shall try these **`Location`**`@serviceLocation` attributes in the order in which they were prioritized in the last steering server response.
    c. As it switches away from the highest priority **`Location`**`@serviceLocation` for local performance reasons, it shall exclude that `@serviceLocation` for a time-limited period equal to the last steering server TTL that it received. Effectively this means that if the next steering server response again assigns the excluded `@serviceLocation` as the highest priority, the client shall ignore that instruction and instead process the `PATHWAY-PRIORITY` array as if the excluded `@serviceLocation` were not present.

8 Extended HTTP GET request parametrization instructions

8.1 URL Query information for Content Steering

Query arguments attached to the request if the DCSM can be used to initialize content steering parameters.

In order to do so, it is recommended that the URL substitution mechanism and syntax as defined in MPEG-DASH ISO/IEC 23009-1 [2], clause I.3, “Extended HTTP GET request parametrization” is used. In order to fully support functionality, the substitution mechanism needs to be extended. Details of this is provided in clause 8.2.

The following provides guidelines on how to make use of this functionality.

Consider a MPD URL of the following form

```
https://cdn.distributor.com/content/common-cachable-
manifest.mpd?steeringToken=12345
```

with relevant contents shown as:

```
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd" type="dynamic"
minimumUpdatePeriod="PT30S" timeShiftBufferDepth="PT30M" availabilityStartTime="2022-02-
25T12:30:00" minBufferTime="PT4S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
     <EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
   xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
     <up:ExtUrlQueryInfo queryTemplate="token=$query:steeringToken$" useMPDUrlQuery="true"
   includeInRequests="steering"/> </up:ExtUrlQueryInfo>
```

```
    <BaseURL serviceLocation="alpha">https://cdn1.example.com/</BaseURL>
    <BaseURL serviceLocation="beta">https://cdn2.example.com/</BaseURL>
    <Period id="1">
      ...
    </Period>
    <ContentSteering defaultServiceLocation="alpha"
  queryBeforeStart="false">https://steeringservice.com/app/instance1234</ContentSteering>
</MPD>
```

In this example the request to the steering server would be processed as

```
https://steeringservice.com/app/instance1234?steeringToken=12345&_DASH_pathway=alpha&_DASH_throughput=5140000
```

8.2 Updates to Annex I - Flexible Insertion of URL Parameters

8.2.1 Introduction

In order to fully support the functionality of content steering, extensions to ISO/IEC 23009-1 [2], clause I.3, "Extended HTTP GET request parametrization" are needed. These documented in the following and communication with MPEG is initiated to address this.

8.2.2 Updates to Table I.3 - description

Table I.3 of ISO/IEC 23009-1 [2] specifies which HTTP `GET` requests are required to carry the URL Parameters. Value is a white spaced concatenated list with keys. In order to support the functionality, a key for content steering needs to be provided, e.g.

7) " steering" (requests to Content Steering servers)

8.2.3 Updates to I.2.4.5

The intent is to re-use the URL parameters of the MPD URL in the Content Steering URL.

Assuming the DASH MPD is accessible through:

```
http://www.example.com/dash/urlparam1.mpd?token=1234&sessionID=h48djn
```

Then

1) Computation of an initial query string

```
initialQueryString="token=1234&sessionID=h48djn"
```

2) Computation of a final query

```
finalQueryString="token=1234&sessionID=h48djn"
```

and the corresponding MPD looks as follows:

```
<MPD
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns="urn:mpeg:dash:schema:mpd:2011"
   xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd
   urn:mpeg:dash:schema:urlparam:2014 DASH-MPD-UP.xsd"
   type="static" mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
   profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
   <EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2014"
   xmlns:up="urn:mpeg:dash:schema:urlparam:2014">
        <up:UrlQueryInfo includeInRequests="7" queryTemplate="$querypart$" useMPDUrlQuery="true"/>
   </EssentialProperty>
   <Period>
      ...
   </Period>
   <ContentSteering defaultServiceLocation="beta" queryBeforeStart="true">
   https://steeringservice.com/app/instance1234</ContentSteering>
</MPD>
```

8.2.4 Modified content steering server URLs building process

In order to create modified content steering server URLs for example as

```
https://steeringservice.com/app/instance1234?token=1234&sessionID=h48djn&_DAS
H_pathway=beta&_DASH_throughput=5140000
```

the _DASH_ parameters are added automatically by the player as part of the processing rules for Content Steering according to clause 7 and are independent of the URL building mechanisms described in Annex I of ISO/IEC 23009-1 [2].

Annex A: Example implementations (informative)

A.1 Basic workflow example

This case illustrates service location changes along with performance override.

A DASH MPD is presented to a player.

```
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd" type="dynamic"
minimumUpdatePeriod="PT30S" timeShiftBufferDepth="PT30M" availabilityStartTime="2022-02-
25T12:30:00" minBufferTime="PT4S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
     <BaseURL serviceLocation="alpha">https://cdn1.com/</BaseURL>
     <BaseURL serviceLocation="beta">https://cdn2.com/</BaseURL>
     <Period id="1">
       <AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F" frameRate="30000/1001"
   segmentAlignment="true" startWithSAP="1">
         <BaseURL>video/</BaseURL>
         …
       </AdaptationSet>
     </Period>
     <ContentSteering defaultServiceLocation="beta"
   queryBeforeStart="true">https://steeringservice.com/app/instance1234?token=234523452</ContentSt
   eering>
</MPD>
```

The player would parse the **`ContentSteering`** element upon receiving the MPD. Since the `@queryBeforeStart` attribute is present and set to `true`, instead of starting playback using the `@defaultServiceLocation` of `"beta"`, it would make a request to the steering server at `https://steeringservice.com/app/instance12345`. This request would be

https://steeringservice.com/app/instance1234?token=234523452

Note the `_DASH_` params are not attached to this request since the player has not yet started playback. The server may then return the JSON response below:

```
{
   "VERSION": 1,
   "TTL": 300,
   "RELOAD-URI": "https://steeringservice.com/app/instance12345?session=abc"
   "PATHWAY-PRIORITY": ["alpha","beta"]
}
```

The player would recognize that the highest priority serviceLocation specified is `"alpha"`, so it would use the **`BaseURL`** construct of `https://cdn1.com/` as it begins to request content. The player would then set a timer so that in 300s, when the throughput it was estimating is 5.14Mbps, it would again query the steering server, with the URL

https://steeringservice.com/app/instance12345?session=abc&_DASH_pathway=alpha&_DASH_throughput=5140000

At that time the steering server may return

```
{
   "VERSION": 1,
   "TTL": 250,
   "RELOAD-URI": "https://steeringservice.com/app/instance12345?session=abc"
   "PATHWAY-PRIORITY": ["beta","alpha"]
}
```

The player would then switch to loading the next media objects using the **`BaseURL`** of `https://cdn2.com/`. 250s later it would again request the steering service and the cycle would continue until end-of-stream was reached.

Let's now assume that the player runs in to a delivery problem 100s after the last steering server response. This problem may be triggered by 404 responses, or throughput degradation. The player decides that `"beta"` is not a good source and makes a local decision to switch to the next highest priority `@serviceLocation "alpha"` and to blacklist `"beta"`. This blacklist last for a time-period equal to the last TTL received, which is 250s. 150s after taking this

action, the player calls the steering server and reports the `@serviceLocation` it is currently playing using the `_DASH_pathway` parameter:

https://steeringservice.com/app/instance12345?session=abc&_DASH_pathway=alpha&_DASH_throughput=4880000

Since the steering server is stateful, it knows that it last assigned `"beta"` but the player is now reporting `"alpha"` implying a client-initiated change. The steering server can take this signal in to account when making its steering decisions. It may however still reply with

```
{
   "VERSION": 1,
   "TTL": 250,
   "RELOAD-URI": "https://steeringservice.com/app/instance12345?session=abc"
   "PATHWAY-PRIORITY": ["beta","alpha"]
}
```

Since the client has excluded `"beta"` for performance reasons for 250s since the switch was made, it processes the `PATHWAY-PRIORITY` array as if `"beta"` were not present and continues to play `"alpha"`. At the next steering server response, the exclusion would have expired and the client should apply the conventional processing rules to the response.

A.2 Advanced steering example

This case illustrates the independent steering of manifests, media segments and advertising content.

A DASH MPD is presented to a player.

```
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd" type="dynamic"
minimumUpdatePeriod="PT30S" timeShiftBufferDepth="PT30M" availabilityStartTime="2022-02-
25T12:30:00" minBufferTime="PT4S" profiles="urn:mpeg:dash:profile:isoff-live:2011">
     <Location serviceLocation="1234">https://manifest-cdn1.com/</Location>
     <Location serviceLocation="5678">https://manifest-cdn2.com/</Location>
     <BaseURL serviceLocation="alpha">https://segments-cdn-A.com/</BaseURL>
     <BaseURL serviceLocation="beta">https://segments-cdn-B.com/</BaseURL>
     <Period id="Primary-Content-1">
      …
     </Period>
     <Period id="Ad-break-1">
        <BaseURL serviceLocation="ad1">https://ad-server-1.com/</BaseURL>
        <BaseURL serviceLocation="ad2">https://ad-server-2.com/</BaseURL>
        …
     </Period>
     <Period id="Primary-Content-2">
         <BaseURL serviceLocation="gamma">https://segments-cdn-C.com/</BaseURL>
         <BaseURL serviceLocation="delta">https://segments-cdn-D.com/</BaseURL>
          …
     </Period>
     <Period id="Ad-break-2">
        <BaseURL serviceLocation="ad3">https://ad-server-3.com/</BaseURL>
        <BaseURL serviceLocation="ad4">https://ad-server-4.com/</BaseURL>
         …
     </Period>
     <Period id="Primary-Content-3">
      …
     </Period>
     <ContentSteering defaultServiceLocation="1234,alpha,ad1">
         https://steeringservice.com/app?token=567
     </ContentSteering>
</MPD>
```

Since **`ContentSteering`**`@queryBeforeStart` is not TRUE, the player would start by matching the content of the **`ContentSteering`**`@defaultServiceLocation` elements against the available **`Location`** and **`BaseURL`** options. As a result, it would start with

1. Load media segments within period “primary-content-1” from https://segments-cdn-A.com.
2. Refresh the manifest from https://manifest-cdn1.com/</

As soon as playback has started and the player’s target buffer has been reached, it would make the following request to the content steering server:

```
https://steeringservice.com/app?token=567&_DASH_pathway="1234,alpha"&_DASH_th
roughput=32000000,19000000
```

Assume the steering server then returns the following JSON response for all future refreshes:

```
{
   "VERSION": 1,
   "TTL": 300,
   "RELOAD-URI": "https://steeringservice.com/app/instance1234"
   "PATHWAY-PRIORITY":
["beta","alpha","ad1","delta","5678","gamma","1234","ad4","ad3"
]
}
```

Following the rules of selecting the highest matching **`Location`** and **`BaseURL`** serviceLocation from the PATHWAY-PRIORITY array, along with **`BaseURL`** inheritance, the client would:

3. Switch to loading media segments within period “primary-content-1” from https://segments-cdn-B.com.
4. Switch to loading the manifest from https://manifest-cdn2.com/</
5. Load media segments within period “ad-break-1” from https://ad-server-1.com/</
6. Load media segments within period “primary-content-2” from https://segments-cdn-D.com
7. Load media segments within period “ad-break-2” from https://ad-server-4.com
8. Load media segments within period “primary-content-3” from https://segments-cdn-B.com

Assuming that the second steering server request (after 300s) occurs during period “Primary-Content-2”, then client would construct the request in the following manner:

```
https://steeringservice.com/app/instance1234?_DASH_pathway="5678,beta,ad2,gam
ma"&_DASH_throughput=450000,56000000,21000000,32000000
```

Annex (informative): Change History

Date	Version	Information about changes
2022-07-10	0.9.0	Version published for community review.
2022-12-07	0.9.5	Updates following community review.
2022-12-21	0.9.6	Version submitted for second community review