

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 April 2026

G. Bichot
Broadpeak
A. Sioniz
Telefonica
G. Goldstein
17 October 2025

CDNI Delivery Metadata
draft-ietf-cdni-delivery-metadata-00

Abstract

This specification adds to the core set of configuration metadata defined in RFC8006, providing delivery metadata to define traffic types, request delegation behavior for downstream CDN (dCDN) node selection, and request routing modes of traffic delegation.

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 20 April 2026.

Copyright Notice

Copyright (c) 2025 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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Cdn Selection	2
1.2. Request Routing	3
1.3. Traffic Characterization	3
2. Requirements	4
3. MI.CdnSelection	4
4. MI.RequestRouting	5
5. MI.TrafficType	6
6. MI.MediaServiceDescription	7
7. Capabilities Advertisements	9
7.1. FCI.CdnSelection	9
7.1.1. FCI.CdnTransport	9
8. Security Considerations	11
9. IANA Considerations	11
9.1. CDNI Payload Types	11
10. Acknowledgements	12
11. Normative References	12
12. Informative References	12
Authors' Addresses	12

1. Introduction

This specification introduces a set of metadata objects and related footprint and capabilities objects that guide content delivery. The specification includes traffic characterization, downstream CDN (dCDN) selection directives, and request routing related metadata.

Without any specific instruction, those metadata objects defined hereafter apply to the configuration associated with a host match and possibly a path match according to [RFC8006].

1.1. Cdn Selection

The downstream CDN is by default viewed as a unique delivery infrastructure. There are however situations and implementation cases where the delivery infrastructure may differ depending on the content type like real-time (i.e. very low latency) content, WEB/file download, multicast (vs unicast) delivery, etc. This document introduces the principle of multiple downstream CDN transport arrangements. A downstream CDN transport arrangement is associated with supported traffic types, capacity limits, transport type (multicast versus unicast).

There exists by default one defacto/implicit downstream CDN arrangement. Several downstream CDN transport arrangements can be exposed through dedicated new Footprint and Capabilities (FCI) objects and conversely be exploited by uCDN using new configuration objects depicted in this document.

Depending on a CDN selection policy indicated by the uCDN, the dCDN may attempt to select the preferred dCDN transport arrangement and if not possible (e.g. no more capacity) may indicate an error or may select a backup virtual dCDN.

1.2. Request Routing

Iterative CDNI Request Redirection is defined in Section 1.1 of [RFC7336] and elaborated by examples in Sections 3.2 and 3.4 of [RFC7336]. Footprint and capabilities for redirection modes are exposed by the dCDN through FCI objects according to [RFC8008]. This includes Iterative redirection modes based on the Domain Name System (DNS) or HTTP redirect. Supporting the mode "I-HTTP" means for the dCDN that it can perform request routing based on that method.

There are examples missing in [RFC7336] that is the possibility to mix request routing methods. Coming back to the examples disclosed in sections 3.2 (Iterative HTTP Redirect example) and 3.4 (Iterative DNS-based redirection example) of [RFC7336], nothing precludes the dCDN (i.e. operator B in the examples) to operate DNS based request routing and HTTP redirect request routing respectively. Therefore, when the uCDN operates one particular iterative redirection mode, routing the request to the dCDN, there is no guarantee about what request routing method the dCDN will use.

The uCDN requires the ability to indicate the preferred iterative request routing method among those supported by the dCDN. This is REQUIRED in cases where the uCDN would like to delegate the traffic relying on the iterative method but knows the client will not support for instance HTTP redirection. In that case, the uCDN needs a means to force the dCDN to perform request routing based on e.g. DNS redirect instead of HTTP redirect.

1.3. Traffic Characterization

Content delivery networks often apply different infrastructure, network routes, and internal metadata for different types of traffic. Delivery of large static objects (such as software downloads), for example, may use different edge servers and network routes than video stream delivery. In an HTTP adaptive bitrate video service, every video title corresponds to a set of video files and descriptors according to different video protocols, and this is independent of

the type of service (video-on-demand, live, catch-up, etc.).

The way the video service is consumed by the user agents can vary. For instance, a segment that belongs to a video on demand (VOD) title can be requested for every moment the content is available for the user agents to consume, while a segment of live content will be only requested as long as the time-shift duration is configured for that service. Knowing those differences, a provider can implement specific strategies that will maximize performance and thereby provide more available capacity to the upstream provider. It should be noted that the dCDNs handling of the traffic types is implementation-specific and not prescribed here.

2. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. MI.CdnSelection

MI.CdnSelection is a GenericMetadata object that allows the uCDN to indicate a preference to one of multiple dCDN transport arrangements..

Configuration metadata is required to select the dCDN arrangement among a set of possibilities exposed by the dCDN through the Footprint and Capability Interface (the FCI.CdnSelection object). Associated with the dCDN delivery network, there is a default selection policy that is "best-effort", i.e., the dCDN tries its best to fulfill the requested policy without providing guarantees.

Property: cdn-name

- * Description: Instructs the dCDN to perform delegation operations for a particular dCDN transport arrangement. The MI.CdnSelection configuration object MUST be associated with the metadata configuration objects dedicated to a particular host or a particular path match (see [RFC8006])
- * Type: String. Must be one of those dCDN transport arrangement names as exposed by the dCDN through the FCI.CdnTransportSelection object.
- * Mandatory-to-Specify: Yes.

Property: cdn-transport-selection

- * Description: Enforces the selection of this downstream CDN (dCDN) transport arrangement according to a specified policy.
- * Type: String. One of "attempt-or-failed", "attempt-or-besteffort", or "best-effort". For either of the first two values, the delegation MUST be attempted according to the dCDN delivery arrangement corresponding to the cdn-name property . If this is not possible, it is considered as an error and either fails (configuration failure) or the dCDN continues with a "best-effort" procedure respectively.. The "best-effort" value instructs the dCDN to try its best to fulfill the requested cdn-selection policy with no guarantees..
- * Mandatory-to-Specify: No. The value "best-effort" is the default selection policy.

The following is an example of the MI.CdnSelection object:

```
{
  "generic-metadata-type": "MI.CdnSelection",
  "generic-metadata-value": {
    "cdn-name": "MULTICAST",
    "cdn-selection": "attempt-or-failed"
  }
}
```

Figure 1

4. MI.RequestRouting

MI.RequestRouting is a GenericMetadata object that allows the uCDN to force the dCDN request routing method(s) to be applied when working in iterative redirection mode.

Property: request-routing-modes

- * Description: Instructs the dCDN to perform request routing according to one or more preferred modes among those supported and advertised by the dCDN through the FCI.RequestRouting object. One must understand that forcing (instead of letting the dCDN request router select) one particular request routing mode may trigger some inefficiency in the request routing process.
- * Type: Array of iterative request routing modes. The values are: "DNS", "HTTP", or "MANIFEST_REWRITE".

- * **Mandatory-to-Specify:** No. By default, all request routing modes supported by the dCDN can be used by the dCDN as part of its request routing process.

The following example, illustrates the uCDN forcing the dCDN to use DNS or HTTP as the method for request routing in case the uCDN performs an iterative delegation (i.e., iterative redirection mode):

```
{
  "generic-metadata-type": "MI.RequestRouting",
  "generic-metadata-value": {
    "request-routing-modes": [ "DNS", "HTTP" ]
  }
}
```

Figure 2

5. MI.TrafficType

MI.TrafficType metadata defines a set of descriptors that characterize either the type or usage of the traffic, enabling providers to apply any internal configuration rules without exposing an unnecessary number of internal details. Note that the interpretation of these traffic types and application of rules, such as rate limiting or delivery pacing, are implementation specific.

The metadata object applies to the configuration associated with a host match and possibly a path match according to [RFC8006]. It is possible to declare several MI.TrafficType objects indicating the support of several traffic types.

Property: traffic-type

- * **Description:** Designates the traffic type. The uCDN will use the literal that is most representative of the traffic being delegated.
- * **Type:** String, one of (vod | live | object-download)
 - vod: This is streaming of on demand video. The delivery can be rate controlled.
 - live: this is streaming live with possible constraints in terms of latency.
 - object-download: This is typically file download like WEB objects or VOD files. The delivery can be rate controlled.

* Mandatory-to-Specify: Yes

Property: hints

* Description: Other traffic characteristics that the uCDN can indicate to the dCDN as suggestions for service optimization. Some other specifications may impose some well-defined values [SVTA2065]

* Type: Array of strings

* Mandatory-to-Specify: No

The following is an example of MI.TrafficType that designates VOD catch-up TV viewing:

```
{
  "generic-metadata-type": "MI.TrafficType",
  "generic-metadata-value": {
    "traffic-type": "vod",
    "hints": [ "catch-up" ]
  }
}
```

Figure 3

6. MI.MediaServiceDescription

MI.MediaServiceDescription metadata defines a set of descriptors associated with a media service delegated to the dCDN. The metadata object applies to the configuration associated with a host match and possibly a path match according to [RFC8006]. A Metadata set associated with a host match or path match SHALL NOT gather more than one MediaServiceDescription metadata object.

The uCDN MAY use this metadata object for controlling the bandwidth budget (possibly previously allocated by the dCDN) obtained through the Capacity Insight Interface [SVTA2049] or possibly offline.

This metadata can be used by the dCDN provider to implement specific strategies that will maximize performance. Note that these strategies are implementation specific and not specified in this document. With knowledge of the streaming manifest URL, for example, the dCDN MAY implement segment prefetching strategies. Furthermore, the notion of a media service MAY allow the dCDN provider to track and monitor streaming sessions in a more comprehensive manner.

Property: manifestURL

- * Description: Path of the manifest (e.g. mpd or m3u8) file related to this media service.
- * Type: String.
- * Mandatory-to-Specify: No.

Property: mediaServiceName

- * Description: String describing or identifying the media service. This MAY be used by the uCDN to correlate several configuration metadata objects with a name (e.g. a TV channel name)
- * Type: String.
- * Mandatory-to-Specify: No.

Property: maximumBitrate

- * Description: This is the maximum bitrate in bits per second (bps) attached to the service delivery. If the service includes multiple representations or playlists, this property restricts the bitrate of each representation or playlist with a published bitrate to a value below this property's value. The property's value indicates the maximum bitrate provisioned for the service, regardless of the representations or playlists. This property must be set according to the maximum bitrate dedicated to the uCDN by the dCDN and published through the FCI.CdnDelivery (limit type "ingress") and/or the FCI.CapacityLimit (limit type "ingress").
- * Type: integer.
- * Mandatory-to-Specify: No. If not specified, the uCDN relies entirely on the dCDN for multiple services delivery. It is strongly encouraged to specify a maximum bitrate for allowing the uCDN to operate multicast delivery for several concurrent serv.

The following example of MI.MediaServiceDescription pointing to the manifest of a live channel and associates a name to this channel:


```
{
  "generic-metadata-type": "MI.MediaServiceDescription",
  "generic-metadata-value": {
    "manifestURL": "/live/channelXYZ/index.mpd",
    "mediaServiceName": "ChannelXYZ",
    "maximumBitRate": 5000000,
  }
}
```

Figure 4

7. Capabilities Advertisements

This section introduces FCI objects that allow a dCDN to advertise its specific capabilities related to delivery.

7.1. FCI.CdnSelection

This object is used by the dCDN to advertise the supported CDN transport arrangements.

Property: `cdn-transport-list`

- * Description: A list of supported dCDN network transport arrangements.
- * Type: Array of FCI.CdnTransport objects that specify the allowed combinations.
- * Mandatory-to-Specify: Yes.

7.1.1. FCI.CdnTransport

This sub-object is used to describe a particular dCDN transport arrangement.

Property: `cdn-name`

- * Description: name of the downstream CDN transport arrangement. A downstream CDN operator may own several transport arrangements which may be the subject of a different delegation. A dCDN transport arrangement is named according to that property. The name is used with the corresponding MI.CdnSelection for pointing out one particular transport arrangement associated with the configuration metadata.
- * Type: String.

- * **Mandatory-to-Specify:** Yes. The name **MUST** be unique among the list exposed by the `cdn-transport-list` property of the `FCI.CdnSelection` object.

Property: `cdn-transport-type`

- * **Description:** Transport type of the downstream CDN transport arrangement..
- * **Type:** String. Must be one of these values: "MULTICAST", "UNICAST".
- * **Mandatory-to-Specify:** No. The default is unicast. There is always one default dCDN unicast transport arrangement.

Property: `cdn-capacity-limits`

- * **Description:** Exposes some capacity limits associated with that dCDN transport arrangement
- * **Type:** An array of `FCI.CapacityLimit` objects as defined in [SVTA2049].
- * **Mandatory-to-Specify:** No.

Property: `cdn-traffic-types`

- * **Description:** a set of traffic types supported by this dCDN transport arrangement.
- * **Type:** An array of `MI.TrafficType` objects
- * **Mandatory-to-Specify:** No.

The following is an example advertising support for Multicast delivery:

```

{
  "capabilities": [
    {
      "capability-type": "FCI.CdnSelection",
      "capability-value": {
        "cdn-transport-list": [
          {
            "cdn-name": "MULTICAST",
            "cdn-transport-type": "MULTICAST",
            "cdn-capacity-limit": [
              {
                "id": "capacity_limit_multicast",
                "limit-type": "ingress",
                "maximum-hard": 50000000,
                "maximum-soft": 40000000,
                "current": 15000000
              },
              {
                "id": "capacity_limit_multicast",
                "limit-type": "egress",
                "maximum-hard": 5000000,
                "maximum-soft": 4000000
              }
            ],
            "cdn-traffic-types": [
              {
                "traffic-type": "live"
              }
            ]
          }
        ]
      }
    }
  ]
}

```

Figure 5

8. Security Considerations

9. IANA Considerations

9.1. CDNI Payload Types

TBD.

10. Acknowledgements

The authors would like to express their gratitude to the members of the Streaming Video Technology Alliance [SVTA] Open Caching Working Group for their guidance / contribution / reviews ...)

Particularly the following people contribute in one or other way to the content of this draft:

- * Person 1 (Company 1)

11. Normative References

- [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>>.
- [RFC8006] Niven-Jenkins, B., Murray, R., Caulfield, M., and K. Ma, "Content Delivery Network Interconnection (CDNI) Metadata", RFC 8006, DOI 10.17487/RFC8006, December 2016, <<https://www.rfc-editor.org/info/rfc8006>>.
- [RFC8008] Seedorf, J., Peterson, J., Previdi, S., van Brandenburg, R., and K. Ma, "Content Delivery Network Interconnection (CDNI) Request Routing: Footprint and Capabilities Semantics", RFC 8008, DOI 10.17487/RFC8008, December 2016, <<https://www.rfc-editor.org/info/rfc8008>>.

12. Informative References

- [RFC7336] Peterson, L., Davie, B., and R. van Brandenburg, Ed., "Framework for Content Distribution Network Interconnection (CDNI)", RFC 7336, DOI 10.17487/RFC7336, August 2014, <<https://www.rfc-editor.org/info/rfc7336>>.
- [SVTA] "Streaming Video Technology Alliance Home Page", <<https://www.svta.org>>.
- [SVTA2049] "Capacity Insights Interface", <<https://svta.org/documents/SVTA2049>>>.
- [SVTA2065] "SVTA Open Casting", <<https://svta.org/documents/SVTA2065>>>.

Authors' Addresses

Guillaume Bichot
Broadpeak
France
Email: guillaume.bichot@broadpeak.tv

Alfonso Soloniz
Telefonica
Spain
Email: alfonso.siloniz.ext@telefonica.com

Glenn Goldstein
United States of America
Email: glennng1215@gmail.com