

Content Delivery Networks Interconnection  
Internet-Draft  
Intended status: Standards Track  
Expires: 28 February 2026

G. Bichot  
Broadpeak  
A. Sioniz  
Telefonica  
G. Goldstein  
Lumen Technologies  
27 August 2025

CDNI Delivery Metadata  
draft-bichot-delivery-metadata-05

## 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 28 February 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
2. Requirements . . . . .	2
3. MI.CdnSelection . . . . .	2
4. MI.RequestRouting . . . . .	4
5. MI.TrafficType . . . . .	5
6. MI.MediaServiceDescription . . . . .	7
7. Capabilities Advertisements . . . . .	8
7.1. FCI.RequestRouting . . . . .	8
7.2. FCI.CdnSelection . . . . .	9
7.2.1. FCI.CdnDelivery . . . . .	9
8. Security Considerations . . . . .	11
9. IANA Considerations . . . . .	12
9.1. CDNI Payload Types . . . . .	12
10. Acknowledgements . . . . .	12
11. Normative References . . . . .	13
12. Informative References . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

This specification introduces a set of metadata objects and related footprint and capabilities objects that guide content delivery. The specification includes traffic types and service descriptions, dCDN node selection directives, and request routing metadata.

## 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 the dCDN, in terms of dCDN delivery arrangement selection.

Configuration metadata is required to select the dCDN delivery networks and/or transport arrangements 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.

For more details on Open Casting node selection, refer to the Open Casting specification [SVTA2065] and Open Casting Request Routing Functional Specification [SVTA2007]

Property: cdn-name

- \* Description: Instructs the dCDN to perform delegation operations for a particular delivery 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 delivery arrangement names as exposed by the dCDN through the FCI.CdnSelection object.
- \* Mandatory-to-Specify: Yes.

Property: cdn-selection

- \* Description: Enforces the selection of this downstream CDN (dCDN) delivery arrangement according to a specified policy.
- \* Type: String. One of "attempt-or-failed", "attempt-or-besteffect", 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. The "best-effort" value instructs the dCDN to try its best to fulfill the requested cdn-selection policy with no guarantees. See the Open Casting specification [SVTA2065] for more explanation related to the multicast delivery scenario.
- \* 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

The uCDN requires the ability to indicate whether Hypertext Transfer Protocol (HTTP) redirect, Domain Name System (DNS) redirect, and manifest rewrite are allowed, and indicate which is preferable. 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 HTTP redirection. In that case, the uCDN needs a means to force the dCDN to perform request routing based on DNS redirect (or manifest rewrite).

For more details on Open Caching request routing, refer to the Open Caching Request Routing Functional Specification [SVTA2007]

This configuration possibility is useful only if the dCDN can advertise the mode of redirection it supports. There is an ongoing discussion in the IETF CDNI group to understand the semantics behind the redirection modes currently in the Footprint & Capabilities Advertising Interface (I-DNS and I-HTTP). It is not clear whether this indicates that the dCDN supports one or both delegation modes (the request routing performed by the uCDN can only be based on DNS redirect or HTTP redirect, or both), or whether it indicates that the dCDN supports, as its own request routing mode, DNS redirect and/or HTTP redirect. The latter is REQUIRED for this new configuration object to be valid.

MI.RequestRouting is a GenericMetadata object that allows the uCDN to force the dCDN request routing mode(s) to be applied when working in iterative redirection mode. The list of redirection modes supported by the dCDN is advertised through the FCI.RedirectionMode object. The list of request routing modes supported by the dCDN is advertised through the FCI.RequestRoutingMode object documented in the Capabilities Advertisements (Section 7) section.

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

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), may, for example, 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.

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.

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)
- \* Mandatory-to-Specify: Yes

Property: hints

- \* Description: Other traffic characteristics that the uCDN can indicate to the dCDN as suggestions for service optimization. This property accepts by default free-form unconstrained values. Some SVTA specifications may impose some well-defined values, as with the Open Casting specification [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. 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 (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.
- \* 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. In multicast, the property's value indicates the maximum bitrate provisioned for the service, regardless of the representations or playlists sent concurrently. This property must be set according to the maximum bitrate dedicated to the uCDN by the dCDN and published through the FCI.CdnSelection (limit type "ingress"). In unicast, the property's value indicates the maximum bitrate provisioned for the service regardless of the representation/playlist that is downstream. This property must be set according to the maximum bitrate dedicated to the uCDN by the dCDN and published through FCI.CapacityLimit (limit type "ingress") or possibly through the FCI.CdnSelectionMulticastDelivery (limit type "ingress").

- \* Type: integer.
- \* Mandatory-to-Specify: No. If not specified, the uCDN relies entirely on the dCDN for all uCDN multicast service delivery. It is strongly encouraged to specify a maximum bitrate for allowing the uCDN to operate multicast delivery for several concurrent services involving live delivery.

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 the MI.RequestRouting and MI.CdnSelection and objects.

### 7.1. FCI.RequestRouting

This object is used by the dCDN to advertise the supported request routing modes. This can be optionally used by the uCDN to further select a subset of those modes when operating one of the iterative delegation modes. See the section MI.RequestRouting (Section 4)

Property: request-routing-modes

- \* Description: A list of supported request routing modes by the dCDN. This information is useful when the uCDN decides to perform a delegation in iterative mode.
- \* Type: Array of strings. Values are: "DNS", "HTTP-R", or "MANIFEST\_REWRITE".
- \* Mandatory-to-Specify: No. If the dCDN does not advertise the supported request routing modes, they are all supported by default.



The following example advertises support for all the request routing modes:

```
{
  "capabilities": [
    {
      "capability-type": "FCI.RequestRouting",
      "capability-value": {
        "request-routing-modes": [
          "DNS",
          "HTTP",
          "MANIFEST_REWRITE"
        ]
      }
    }
  ]
}
```

Figure 5

## 7.2. FCI.CdnSelection

This object is used by the dCDN to advertise the network delivery arrangements supported by the dCDN.

Property: cdn-delivery-list

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

### 7.2.1. FCI.CdnDelivery

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

Property: cdn-name

- \* Description: name of the downstream CDN delivery arrangement. A downstream CDN operator may own several delivery infrastructure instances which may be the subject of a different delegation. A dCDN delivery arrangement is named according to that property. The name is used with the corresponding MI.CdnSelection for pointing out one particular delivery arrangement associated with the configuration metadata.
- \* Type: String.
- \* Mandatory-to-Specify: Yes. The name MUST be unique among the list exposed by the Cdn-delivery-list property of the FCI.CdnSelection object.

Property: cdn-transport-type

- \* Description: Main transport characteristic of the downstream CDN delivery instance.
- \* 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 delivery infrastructure/arrangement.

Property: cdn-capacity-limits

- \* Description: exposes some capacity limits associated with that dCDN delivery arrangement
- \* Type: An array of FCI.CapacityLimit objects as defined in [SVTA2049] and [SVTA2065]
- \* Mandatory-to-Specify: No.

Property: cdn-traffic-types

- \* Description: a set of traffic types supported by this dCDN delivery 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-delivery-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": 50000000,
                "maximum-soft": 40000000
              }
            ],
            "cdn-traffic-types": [
              {
                "traffic-type": "live"
              }
            ]
          }
        ]
      }
    }
  ]
}

```

Figure 6

## 8. Security Considerations

The FCI and MI objects defined in this document are transferred via the interfaces defined in CDNI [RFC8006] which describes how to secure these interfaces by protecting integrity and confidentiality while ensuring the authenticity of the dCDN and uCDN.

## 9. IANA Considerations

### 9.1. CDNI Payload Types

This document requests the registration of the following entries under the "CDNI Payload Types" registry hosted by IANA:

Payload Type	Specification
MI.CdnSelection	RFCthis
MI.RequestRouting	RFCthis
MI.TrafficType	RFCthis
MI.MediaServiceDescription	RFCthis
FCI.CdnSelection	RFCthis
FCI.CdnDelivery	RFCthis

Table 1: CDNI Payload Types

## 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 contributions and guidance.

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

- \* Christoph Neumann - Broadpeak
- \* Will Power - Lumen
- \* Shmuel Asafi - Qwilt
- \* Yoav Gressel - Qwilt
- \* Nir Sopher - Qwilt
- \* Arnon Warshavsky - Qwilt
- \* Francisco Cano Hila - Telefonica

## 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>>.

## 12. Informative References

- [SVTA] SVTA, "Streaming Video Technology Alliance Home Page", <<https://www.svta.org>>.
- [SVTA2007] SVTA, "Open Caching Request Routing Functional Specification", <<https://svta.org/documents/SVTA2007>>.
- [SVTA2049] SVTA, "Capacity Insights Interface", <<https://svta.org/documents/SVTA2049>>.
- [SVTA2065] SVTA, "SVTA Open Casting", <<https://svta.org/documents/SVTA2065>>.

## Authors' Addresses

Guillaume Bichot  
Broadpeak  
France  
Email: [guillaume.bichot@broadpeak.tv](mailto:guillaume.bichot@broadpeak.tv)

Alfonso Siloniz  
Telefonica  
Spain  
Email: [alfonsosiloniz@gmail.com](mailto:alfonsosiloniz@gmail.com)

Glenn Goldstein  
Lumen Technologies  
United States of America  
Email: [glennng1215@gmail.com](mailto:glennng1215@gmail.com)