Content Delivery Networks Interconnection
Internet-Draft
Intended status: Standards Track
Expires: 29 August 2025

G. Bichot
Broadpeak
A. Siloniz
Telefonica
G. Goldstein
Lumen Technologies

25 February 2025

CDNI Delivery Metadata
draft-bichot-delivery-metadata-02

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 29 August 2025.

Copyright Notice

Table of Contents

1. Introduction 2
2. Requirements 2
3. MI.OcnSelection 3
3.1. MI.OcnDelivery 5
4. MI.RequestRouting 6
5. MI.TrafficType 7
6. MI.MediaServiceDescription 8
7. Capabilities Advertisements 10
7.1. FCI.OcnSelection 10
7.2. FCI.RequestRouting 11
7.3. FCI.MulticastDelivery 12
8. Security Considerations 13
9. IANA Considerations 14
9.1. CDNI Payload Types 14
10. Acknowledgements 14
11. Normative References 15
12. Informative References 15
Authors' Addresses 15

1. Introduction

This specification introduces a set of GenericMetadata objects that guide content delivery. This includes traffic types and service descriptions, dCDN node selection directives, and request routing metadata. For background on the Open Caching standards and interactions between upstream content delivery networks (uCDNs) and downstream content delivery networks (dCDNs) that these configuration settings impact, refer to the Open Caching Request Routing Functional Specification [SVTA2007]

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.OcnSelection

Configuration metadata is required to permit several levels of downstream Open Caching node (OCN) selection policies. For example, in a mobile network, several physical locations are possible (i.e., candidates) for hosting the OCN that will take charge in the delegation for the uCDN. This is the case when the cache is virtualized and deployed dynamically. Depending on the OCN selection policy, which may be a cost driver, the dCDN may attempt to favor certain types of caches at the edge, for example.

The available types of OpenCaching nodes are announced by the dCDN using the FCI.OCNSelection object. The possible types of downstream nodes are not predefined and might for instance be linked to the network characteristics such as "EDGE" or "average latency< 10ms" (i.e., the property ocn-type can be one of these values). The dCDN can select the node types using the ocn-type property. The uCDN can also configure the delivery type using the ocn-delivery property, and specify for instance whether to use MABR (multicast ABR) over a satellite link or HAS (HTTP Adaptive Streaming) over a cellular network. The default OCN selection policy is "best-effort", i.e., the dCDN tries its best to fulfill the requested policy without providing guarantees.

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

MI.OcnSelection is a new GenericMetadata object that allows the uCDN to indicate a preference to the dCDN, in terms of OCN selection.

Property: ocn-delivery

* Description: Instructs the dCDN to perform delegation operations for a particular medium and/or a transport arrangement.

* Type: MI.OcnDelivery object

* Mandatory-to-Specify: No. At least one of the two properties, ocn-type or ocn-delivery, MUST be present.

Property: ocn-type

* Description: Instructs the dCDN to perform delegation operations for the type of downstream nodes.

* Type: A string corresponding to one of the node types announced by the dCDN through the CDNI Footprint & Capabilities Interface (FCI). Examples include: "HOME" or "EDGE".

* Mandatory-to-Specify: No. At least one of the two properties, ocn-type or ocn-delivery, MUST be present.

Property: ocn-selection

* Description: Enforces the selection of OCNs, considering the ocn-type and/or the ocn-delivery properties.

* 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 ocn-type and/or the ocn-delivery properties. 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 ocn-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 OCN selection policy. This property makes sense only when dealing with multicast delivery:

MI.OcnDelivery.ocn-medium="SATELLITE" and/or (MI.OcnDelivery.ocn-transport"="MULTICAST" or MI.OcnDelivery.ocn-transport"="MABR")

Property: multicast-mode

* Description: Permits to enforce the dCDN operation mode when dealing with multicast delivery in addition to unicast delivery (ocn-selection = "attempt-or-besteffort").

* Type: String. For more insights, please read the Open Casting specification [SVTA2065].

  - "automatic-popular": this (default) mode makes the dCDN always trying to multicast the most popular content.

  - "on-demand" : this mode makes the selection of live content events to be multicasted controlled by the uCDN through a dedicated API (see .

* Mandatory-to-Specify: No. The value "automatic-popular" is the default multicast mode. This property makes sense only when dealing with multicast delivery:

MI.OcnDelivery.ocn-medium="SATELLITE" and/or (MI.OcnDelivery.ocn-transport"="MULTICAST" or MI.OcnDelivery.ocn-transport"="MABR") and MI.OcnSelection.ocn-selection = "attempt-or-besteffort"

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

```
{
  "generic-metadata-type": "MI.OcnSelection",
  "generic-metadata-value": {
    "ocn-delivery": {
      "ocn-medium": "SATELLITE",
      "ocn-transport": "MABR"
    },
    "ocn-type": "EDGE",
    "ocn-selection": "attempt-or-failed"
  }
}
```

Figure 1

3.1. MI.OcnDelivery

MI.OcnDelivery is a subobject of MI.OcnSelection that provides details on how the delegated content should be delivered.

Property: ocn-medium

* Description: Instructs the dCDN to perform delegation operations for a particular medium.

* Type: String. Must be one of these values: "SATELLITE", "CELLULAR", "BROADBAND", "TERRESTRIAL".

* Mandatory-to-Specify: No. Either the ocn-medium property or the ocn-transport property MUST be present.

Property: ocn-transport

* Description: Instructs the dCDN to perform delegation operations for a particular transport arrangement.

* Type: String. Must be one of these values: "MULTICAST", "MABR" (Multicast Adaptive Bitrate), "UNICAST" or "HAS" (HTTP Adaptive Streaming).

* Mandatory-to-Specify: No. At least one of the two properties (ocn-medium or ocn-transport) MUST be present.

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 new 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 CDN or OCN 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 CDNs and OCNs 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 CDN or OCN 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 CDN or OCN 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 is composed with several representations/playlists then this property bounds (i.e. capes) the usage of all or any of the representation/playlist that has a published bitrate, below this property's value. In multicast, the property's value indicates the maximum bitrate provisioned for the service whatever the representation(s)/playlist(s) that is(are) sent (concurrently). This property must be set according to the maximum bitrate dedicated to the uCDN by the dCDN and published through FCI.MulticastDelivery (limit types "ingress").In unicast, the property's value indicates the maximum bitrate provisioned for the service whatever 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")..

* Type: integer.

* Mandatory-to-Specify: No. If not specified,the uCDN relies entirely on the dCDN for all uCDN multicast services delivery. It is strongly encouraged to specify a maximum bit-rate 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.OcnSelection and MI.RequestRouting objects.

7.1. FCI.OcnSelection

This object is used by the dCDN to advertise the supported OCN types and/or its transport arrangement, and/or the medium supported by OCNs.

Property: ocn-delivery-list

* Description: A list of supported medium and/or transport arrangements.

* Type: Array of MI.OcnDelivery objects that specify the allowed combinations of medium and transport.

* Mandatory-to-Specify: No

Property: ocn-type-list

* Description: A list of supported OCN types.

* Type: Array of strings. Examples include: "HOME" or "EDGE". The possible types are not predefined and can be freely chosen by the dCDN.

* Mandatory-to-Specify: No

The following is an example advertising support for Satellite Multicast Adaptive Bitrate (MABR) OCN delivery:

```
{
  "capabilities": [
    {
      "capability-type": "FCI.OcnSelection",
      "generic-metadata-value": {
        "ocn-delivery-list": [
          {
            "ocn-medium": "SATELLITE",
            "ocn-transport": "MABR"
          }
        ],
        "ocn-type-list": [
          "HOME",
          "EDGE"
        ]
      }
    }
  ]
}
```

Figure 5

7.2. 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 6

7.3. FCI.MulticastDelivery

The presence of a multicast delivery capability is announced through the FCI.OcsSelection object. This FCI.MulticastDelivery object permits to disclose more information about the multicast delivery.

Property: capacity-limits

* Description: An array of FCI.CapacityLimit objects.

* Type: Array

* Mandatory-to-Specify: Yes

Property: traffic-types

* Description: traffic types available

* Type: An array of MI.TrafficType objects

* Mandatory-to-Specify: yes

The following example advertises support for multicast delivery.:

```
{
  "capabilities":
     [
         {
            "capability-type": "FCI.MulticastDelivery",
            "capability-value": {
                "capacity-limits": [
                    {
                        "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
                    }
                ]
                “traffic-types” : [
                    {
                        "traffic-type": "live",
                        "hints": ["catch-up"]
                    }               ]
            },
            "footprints": [
                {
                    "footprint-type": "ipv4cidr",
                    "footprint-value": [
                        "192.0.2.0/24"
                    ]
                }
            ]
         }
     ]
}
```

Figure 7

8. Security Considerations

The FCI and MI objects defined in the 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.OcnSelection	RFCthis
MI.OcnDelivery	RFCthis
MI.RequestRouting	RFCthis
MI.TrafficType	RFCthis
MI.MediaServiceDescription	RFCthis
FCI.OcnSelection	RFCthis
FCI.RequestRouting	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.

Particulary 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>.

[SVTA2065] SVTA, "SVTA Open Casting", <https://svta.org/documents/SVTA2065>.

[SVTA5044] SVTA, "Open Caching Implementation Guidelines", <https://svta.org/documents/SVTA5044>.

Authors' Addresses

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

Alfonso Siloniz
Telefonica
Spain
Email: alfonsosiloniz@gmail.com

Glenn Goldstein
Lumen Technologies
United States of America
Email: glenng1215@gmail.com