

**Content Distribution Network Interconnection (CDNI) Capabilities
Interface
draft-ma-cdni-capabilities-00**

Abstract

The interconnection of Content Distribution Networks (CDNs) is predicated on the ability of downstream CDNs (dCDNs) to handle end-user requests in a functionally equivalent manner to the upstream CDN (uCDN). The uCDN must be able to assess the ability of the dCDN to handle individual requests. A CDN interconnection (CDNI) interface is needed to facilitate the advertisement of capabilities by the dCDN to the uCDN. This document describes an approach to implementing a CDNI capabilities advertisement interface.

Requirements Language

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 [RFC 2119](#) [[RFC2119](#)].

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 <http://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 February 6, 2013.

Copyright Notice

Copyright (c) 2012 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Terminology	4
2.	Capabilities	4
2.1.	Delivery Protocol	6
2.2.	Acquisition Protocol	7
2.3.	Metadata Bundle	9
2.4.	Redirection Mode	10
3.	Capability Advertisement	11
3.1.	Capability Update	12
3.2.	Capability Removal	13
4.	IANA Considerations	14
5.	Security Considerations	14
6.	Acknowledgements	15
7.	Appendix A: Capability Aggregation	15
7.1.	Downstream CDN Aggregation	15
7.2.	Internal Request Router Aggregation	17
7.3.	Internal Capability Aggregation	18
8.	References	20
8.1.	Normative References	20
8.2.	Informative References	20
	Author's Address	20

1. Introduction

The need for capabilities advertisement in CDNI is described in the CDNI requirements document [[I-D.ietf-cdni-requirements](#)]. Requirement REQ-2 describes the need to allow dCDNs to communicate capabilities to the uCDN. The uCDN aggregates the capabilities information for all dCDNs for use in making request routing decisions. Per requirement REQ-3, the uCDN should further use the aggregated capabilities in advertisements to CDNs further upstream. This concept of aggregation can apply to both organizationally different dCDNs (e.g., other CDN providers, or different business units within a larger organization) or logical entities within the same CDN (e.g., using multiple request routers for scalability reasons, to segregate surrogates based on specific protocol support, or to segregate surrogates based on software version or feature level, etc.).

[Appendix A](#) contains more detailed descriptions of the different capabilities management scenarios, but it is important to note that it is the ability of the dCDN to service each request in a functionally equivalent manner as the uCDN that is important, not the physical layout of resources through which it services the request. The aggregation of resource knowledge by the dCDN into a simple set of capabilities which can be advertised to the uCDN enables efficient decision making at each delegation point in the CDN interconnection hierarchy.

It is assumed that an authoritative request router in each CDN will be responsible for aggregating and advertising capabilities information in a dCDN, and receiving and aggregating capabilities information in the uCDN. As there is no centralized CDNI controller defined in the CDNI architecture, the request router seems the most logical place for capabilities aggregation to occur, as it is the request router that needs such information to make delegation decisions. The protocol defined herein may be implemented as part of an entity other than an authoritative request router, but for the purposes of this discussion, the authoritative request router is assumed to be the centralized capabilities aggregation point.

Though there is an obvious need for the ability to exchange and update capability information in real-time, it is assumed that capabilities do not change very often. There is also an assumption that the capabilities are not by themselves useful for making delegation decisions. Capability information is assumed to be input into business logic. It is the business logic which provides the algorithms for delegation decision making. The definition of business logic occurs outside the scope of CDNI and outside the timescale of capability advertisement. It may be the case that the business logic anticipates and reacts to changes in dCDN

Ma

Expires February 6, 2013

[Page 3]

capabilities. However, it may also be the case that business logic is tailored through offline processes as dCDN capabilities change. The capabilities interface is agnostic to the business processes employed by a given uCDN. The capabilities that are advertised over the capabilities interface may be used by the uCDN at its leisure to implement delegation rules. Setting proper defaults in the business logic should prevent any unwanted delegation from occurring when dCDN capabilities change, however, that is beyond the scope of this discussion.

1.1. Terminology

[Ed. note: Insert terminology reference]

2. Capabilities

There is a basic set of capabilities that must be advertised by the dCDN for the uCDN to be able to determine if the dCDN is functionally able to handle a given request. Mandatory capabilities include:

- o Delivery Protocol
- o Acquisition Protocol
- o Metadata Bundle
- o Redirection Mode

The following sections describe each of the capabilities in further detail, however, all of the capabilities can be described using the same general format. A capability object can be described as:

CapabilityObject:

 Name: Identifier for the capability

 Values: List of supported options for the capability

 Footprint: Optional list of footprint restrictions

The list of valid capability options for a given capability will be specific to the given capability type. Though the degenerate case may exist where the range of option values is a single item, it is anticipated that all capability types will have more than one capability option values. For consistency in the model, all capability types are implemented with lists of values. To optimize actions on the entire range of capability option values for a given capability type, the capability option value "ALL" is reserved and MUST be supported by all capability types. For completeness, the capability option value "NONE" is also reserved and MUST be supported

by all capability types. Reserved values SHOULD NOT be combined with any other valid capability option values for a given capability type. Reserved values MUST NOT be combined with each other for a given capability type. The reserved values MUST be given precedence over all other capability option values for a given capability type, when specified in the same capabilities advertisement message.

The footprint restriction list is optional. Footprint restriction lists override in their entirety all previously advertised footprint restriction lists for the specified capability option values for the given capability type. If no footprint restriction list is specified, it SHALL be understood that all of the capability option values specified have global reach, overriding any previous capability advertisements for the specified capability option values. The footprint restriction list supports either IP prefix or ASN values. IP prefix and ASN restrictions MUST NOT be mixed within a given footprint list. In the case of interconnecting private network CDNs, IP prefix or ASN-based footprint advertisement is the likely method for describing the coverage areas.

[Ed. note: Though other methods exist for defining footprints (e.g., GPS coordinates, country/zip/area code, etc.) only IP prefix and ASN are considered here. Need to evaluate use cases for other methods of footprint definition.]

Note: Further optimization of the capabilities object to provide quality information about a given footprint is certainly possible, however, it is not critical to the basic interconnection of CDNs. The ability to transfer quality information in capabilities advertisements may be desirable and is noted here for completeness, however, the specifics of such mechanisms are outside the scope of this document.

Multiple capability objects of the same capability type are allowed within a given capability advertisement message as long as the capability option values do not overlap, i.e., a given capability option value MUST NOT show up in multiple capability objects within a single capability advertisement message. If multiple capability objects for a given capability type exist, those capability objects SHOULD have different footprint restrictions; capability objects of a given capability type with identical footprint restrictions SHOULD be combined.

Note: The examples shown below use an XML representation, however, other representations (e.g., JSON) may also be acceptable.

2.1. Delivery Protocol

The delivery protocol refers to the protocol over which the client has requested content. If a dCDN does not support the protocol requested by the client, then the dCDN is not a viable candidate for delegation. The delivery protocol is specified in the URI scheme (as defined in RTC3986 [[RFC3986](#)]) in the client request URL.

[Ed. note: The proposal to allow only URI schemes in the specification of delivery protocol is somewhat restrictive in that some CDNs may only support a subset of features defined for that protocol (e.g., different HTTP methods, chunking, ranges, etc.). Need to consider a scalable way to define feature bundles without enumerating every feature of every known protocol.]

[Ed. note: The proposal to allow only URI schemes in the specification of delivery protocol is somewhat restrictive in that it does not differentiate between application layer protocols (e.g., HLS, DASH, etc.) which run over top of HTTP. While support for HTTP should be sufficient to support application layer protocols that run over top of HTTP, there is significant interest in being able to optimize the application layer protocols. Need to consider a way to enhance the protocol definition to include application layer protocols, possibly as a separate capability type.]

The delivery protocol capability object MUST support a list of protocols for a given footprint. The delivery protocol capability SHOULD support optional footprint restriction information. The following XML-encoded example shows two lists of protocols with different footprints.


```
<Capabilities>
  <DeliveryProtocols>
    <Protocols>
      <Protocol>HTTP</Protocol>
      <Protocol>RTSP</Protocol>
      <Protocol>MMS</Protocol>
    </Protocols>
  </DeliveryProtocols>
  <DeliveryProtocols>
    <Protocols>
      <Protocol>RTMP</Protocol>
      <Protocol>HTTPS</Protocol>
    </Protocols>
    <Footprint>
      <Prefixes>
        <Prefix>10.1.0.0/16</Prefix>
        <Prefix>10.10.10.0/24</Prefix>
      </Prefixes>
    </Footprint>
  </DeliveryProtocols>
</Capabilities>
```

In the above example, the three protocols HTTP, RTSP, and MMS are supported globally, while the protocols RTMP and HTTPS are only supported in a restricted footprint (in this case, specified by IP address prefix).

A protocol MUST not appear in multiple delivery protocol lists, within a given capability advertisement message. Redefinition of footprint restrictions in subsequent capability advertisement messages MAY occur and SHALL override all previous capability advertisements for the specified delivery protocol.

2.2. Acquisition Protocol

The acquisition protocol refers to the protocol over which the dCDN may acquire content from the uCDN. If a dCDN does not support any of the protocols offered by the uCDN, then the dCDN is not a viable candidate for delegation. The acquisition protocol is disseminated to the dCDN in the URI scheme (as defined in RTC3986 [[RFC3986](#)]) provided by the uCDN via the CDNI Metadata Interface.

[Ed. note: The proposal to allow only URI schemes in the specification of acquisition protocol is somewhat restrictive in that some CDNs may only support a subset of features defined for that protocol (e.g., different HTTP methods, chunking, ranges, etc.). Need to consider a scalable way to define feature bundles without enumerating every feature of every known protocol.]

[Ed. note: The proposal to allow only URI schemes in the specification of acquisition protocol is somewhat restrictive in that it does not differentiate between application layer protocols (e.g., HLS, DASH, etc.) which run over top of HTTP. While support for HTTP should be sufficient to support application layer protocols that run over top of HTTP, there is significant interest in being able to optimize application layer protocols. Need to consider a way to enhance the protocol definition to include application layer protocols, possibly as a separate capability type.]

The acquisition protocol capability object MUST support a list of protocols for a given footprint. The acquisition protocol capability SHOULD support optional footprint restriction information. The following XML-encoded example shows two lists of protocols with different footprints.

```
<Capabilities>
  <AcquisitionProtocols>
    <Protocols>
      <Protocol>HTTP</Protocol>
      <Protocol>FTP</Protocol>
    </Protocols>
  </AcquisitionProtocols>
  <AcquisitionProtocols>
    <Protocols>
      <Protocol>SFTP</Protocol>
      <Protocol>HTTPS</Protocol>
    </Protocols>
    <Footprint>
      <ASNs>
        <ASN>0</ASN>
        <ASN>65535</ASN>
      </ASNs>
    </Footprint>
  </AcquisitionProtocols>
</Capabilities>
```

In the above example, the two protocols HTTP and FTP are supported globally, while the protocols SFTP and HTTPS are only supported in a restricted footprint (in this case, specified by ASN).

A protocol MUST not appear in multiple acquisition protocol lists, within a given capability advertisement message. Redefinition of footprint restrictions in subsequent capability advertisement messages MAY occur and SHALL override all previous capability advertisements for the specified acquisition protocol.

2.3. Metadata Bundle

Metadata bundles are groupings of one or more metadata definitions which the dCDN is capable of understanding. There will be a core set of metadata defined which all CDNI implementations will be required to support, however, future revisions of CDNI may include additional mandatory to implement metadata, and some operators may require the use of vendor specific metadata. If a dCDN is not capable of understanding a piece of metadata which the uCDN feels is mandatory for delivery, then the dCDN is not a viable candidate for delegation. Metadata bundle naming is to be defined in the CDNI Metadata interface.

[Ed. note: Support for individual options within a given piece of metadata within a given bundle (e.g., URL signing method 3, within the CDNI core metadata version 1 bundle) may also need to be advertised, however, this needs to be coordinated with TBD aspects the CDNI Metadata interface specification.]

The metadata bundle capability object MUST support a list of protocols for a given footprint. The metadata bundle capability SHOULD support optional footprint restriction information. The following XML-encoded example shows two lists of bundles with different footprints.

```
<Capabilities>
  <MetadataBundles>
    <Bundles>
      <Bundle>cdni.core.v1</Bundle>
    </Bundles>
  </MetadataBundles>
  <MetadataBundle>
    <Bundles>
      <Bundle>vendor.azuki.has.v1</Bundle>
      <Bundle>private.rw.color.v1</Bundle>
    </Bundles>
    <Footprint>
      <Prefixes>
        <Prefix>10.1.2.0/24</Prefix>
      </Prefixes>
    </Footprint>
  </MetadataBundle>
</Capabilities>
```

In the above example, core version 1 CDNI metadata `cdni.core.v1` is supported globally, while the vendor specific metadata bundles `vendor.azuki.has.v1` and `private.rw.color.v1` are only supported in a restricted footprint (in this case, specified by IP Prefix).

A bundle MUST not appear in multiple metadata bundle lists, within a given capability advertisement message. Redefinition of footprint restrictions in subsequent capability advertisement messages MAY occur and SHALL override all previous capability advertisements for the specified metadata bundle.

2.4. Redirection Mode

The redirection mode refers to the method or method(s) employed by the request routing interface. The CDNI framework [[I-D.ietf-cdni-framework](#)] document describes four possible request routing modes:

- o DNS iterative (DNS-I)
- o DNS recursive (DNS-R)
- o HTTP iterative (HTTP-I)
- o HTTP recursive (HTTP-R)

If a dCDN supports only a specific mode or subset of modes that does not overlap with the modes supported by the uCDN, then the dCDN is not a viable candidate for delegation.

[Ed. note: The list of request routing modes may not be an exhaustive list of request routing modes that will eventually be supported. This needs to be coordinated with TBD aspects the CDNI Request Routing interface specification.]

The redirection mode capability object MUST support a list of redirection modes for a given footprint. The redirection mode capability SHOULD support optional footprint restriction information. The following XML-encoded example shows two lists of modes with different footprints.


```
<Capabilities>
  <RedirectionModes>
    <Modes>
      <Mode>DNS-I</Mode>
      <Mode>HTTP-I</Mode>
    </Modes>
  </RedirectionModes>
  <RedirectionModes>
    <Modes>
      <Mode>DNS-R</Mode>
      <Mode>HTTP-R</Mode>
    </Modes>
    <Footprint>
      <ASNs>
        <ASN>64511</ASN>
      </ASNs>
    </Footprint>
  </RedirectionModes>
</Capabilities>
```

In the above example, iterative redirection is supported globally, while recursive redirection is only supported in a restricted footprint (in this case, specified by ASN).

A mode MUST not appear in multiple redirection mode lists, within a given capability advertisement message. Redefinition of footprint restrictions in subsequent capability advertisement messages MAY occur and SHALL override all previous capability advertisements for the specified redirection mode.

3. Capability Advertisement

The capabilities advertisement protocol is an HTTP-based protocol using the POST and DELETE methods. Capability removal is distinguished from capability advertisement by the HTTP request method. Capability advertisement uses the HTTP POST method, while capability removal uses the HTTP DELETE method. The dCDN request router asynchronously issues capability advertisement messages to the uCDN request router. It is assumed that the dCDN request router has been configured with the uCDN request router address either through the CDNI Control interface bootstrapping function, or through some other out-of-band configuration.

Note: The examples shown below use an XML representation, however, other representations (e.g., JSON) are also acceptable.

3.1. Capability Update

A capabilities advertisement message may combine objects from one or more capability types. On an initial advertisement, the dCDN SHOULD send a value for every capability type.

```
POST /CDNI/RI/capabilities HTTP/1.1
Host: rr0.u.example.com
Accept: */*
Content-Length: 530
Content-Type: application/x-www-form-urlencoded
```

```
<Capabilities>
  <DeliveryProtocols>
    <Protocols>
      <Protocol>HTTP</Protocol>
    </Protocols>
  </DeliveryProtocols>
  <AcquisitionProtocols>
    <Protocols>
      <Protocol>HTTP</Protocol>
      <Protocol>FTP</Protocol>
    </Protocols>
  </AcquisitionProtocols>
  <MetadataBundles>
    <Bundles>
      <Bundle>cdni.core.v1</Bundle>
      <Bundle>private.rw.color.v1</Bundle>
    </Bundles>
  </MetadataBundles>
  <RedirectionModes>
    <Modes>
      <Mode>HTTP-I</Mode>
    </Modes>
  </RedirectionModes>
</Capabilities>
```

In the above example, the dCDN issues a post to the uCDN request router rr0.u.example.com with a capability advertisement message specifying support for HTTP-based delivery, HTTP or FTP-based content acquisition, iterative HTTP redirection, and support for both core and color metadata. A subsequent capabilities advertisement message may be used to update this information without respecifying all the fields. In the following example, the dCDN updates its ability to deliver content via HTTPS while restricting its previously advertised support for delivering content via HTTP to only its internal network. All previous advertisements for acquisition protocol, metadata bundles, and redirection modes still are unaffected.


```
POST /CDNI/RI/capabilities HTTP/1.1
Host: rr0.u.example.com
Accept: */*
Content-Length: 356
Content-Type: application/x-www-form-urlencoded
```

```
<Capabilities>
  <DeliveryProtocols>
    <Protocols>
      <Protocol>HTTPS</Protocol>
    </Protocols>
  </DeliveryProtocols>
  <DeliveryProtocols>
    <Protocols>
      <Protocol>HTTP</Protocol>
    </Protocols>
    <Footprint>
      <Prefixes>
        <Prefix>10.0.0.0/8</Prefix>
      </Prefixes>
    </Footprint>
  </DeliveryProtocols>
</Capabilities>
```

3.2. Capability Removal

In the following example, the dCDN issues a DELETE command to the uCDN request router rr0.u.example.com with a capability advertisement message containing all four capability types specifying the reserved capability option value ALL. This removes capabilities information for all capability options for all capability types, effectively allowing the dCDN to remove itself from delegation consideration.


```
DELETE /CDNI/RI/capabilities HTTP/1.1
Host: rr0.u.example.com
Accept: */*
Content-Length: 442
Content-Type: application/x-www-form-urlencoded
```

```
<Capabilities>
  <DeliveryProtocols>
    <Protocols>
      <Protocol>ALL</Protocol>
    </Protocols>
  </DeliveryProtocols>
  <AcquisitionProtocols>
    <Protocols>
      <Protocol>ALL</Protocol>
    </Protocols>
  </AcquisitionProtocols>
  <MetadataBundles>
    <Bundles>
      <Bundle>ALL</Bundle>
    </Bundles>
  </MetadataBundles>
  <RedirectionModes>
    <Modes>
      <Mode>ALL</Mode>
    </Modes>
  </RedirectionModes>
</Capabilities>
```

4. IANA Considerations

This memo includes no request to IANA.

[Ed. note: Should there be a registry for capability names?]

5. Security Considerations

There are a number of security concerns associated with the capabilities interface. As the capabilities interface essentially provides configuration information which the uCDN uses to make request routing decisions. Injection of fake capability advertisement messages, or interception and discard of real capability advertisement messages may be used for denial of service (e.g., by falsely advertising or deleting capabilities or preventing capability advertisements from reaching the uCDN). dCDN capability advertisements MUST be authenticated by the uCDN to prevent

unauthorized capability advertisement message injection. uCDN request router capability interface servers MUST be authenticated by the dCDN to prevent unauthorized interception of capability advertisement messages. TLS with client authentication SHOULD be used for all capability interface implementations. Deployments in controlled environments where physical security and IP address white-listing is employed MAY choose not to use TLS.

6. Acknowledgements

The authors would like to thank Ray van Brandenburg and Gilles Bertrand for their expeditious reviews and invaluable comments.

7. [Appendix A](#): Capability Aggregation

The following sections show examples of three aggregation scenarios. In each case, CDN-U is the ultimate uCDN and CDN-P is the penultimate CDN which must perform capabilities aggregation.

[7.1](#). Downstream CDN Aggregation

Figure A1 shows five organizationally different CDNs: CDN-U, CDN-P, and CDNs A, B, and C, the dCDNs of CDN-P which are being aggregated. Given the setup shown in Figure A1, we can construct a number of use cases, based on the coverage areas of each dCDN (i.e., CDNs P, A, B, and C). Note: In all cases, the reachability of the uCDN (i.e., CDN-U) is a don't care as it is assumed that the uCDN knows its own coverage area and is likely to favor itself in most situations, and if it has decided that it needs to delegate to a dCDN, then the only relevant question is if the dCDN can handle the request.

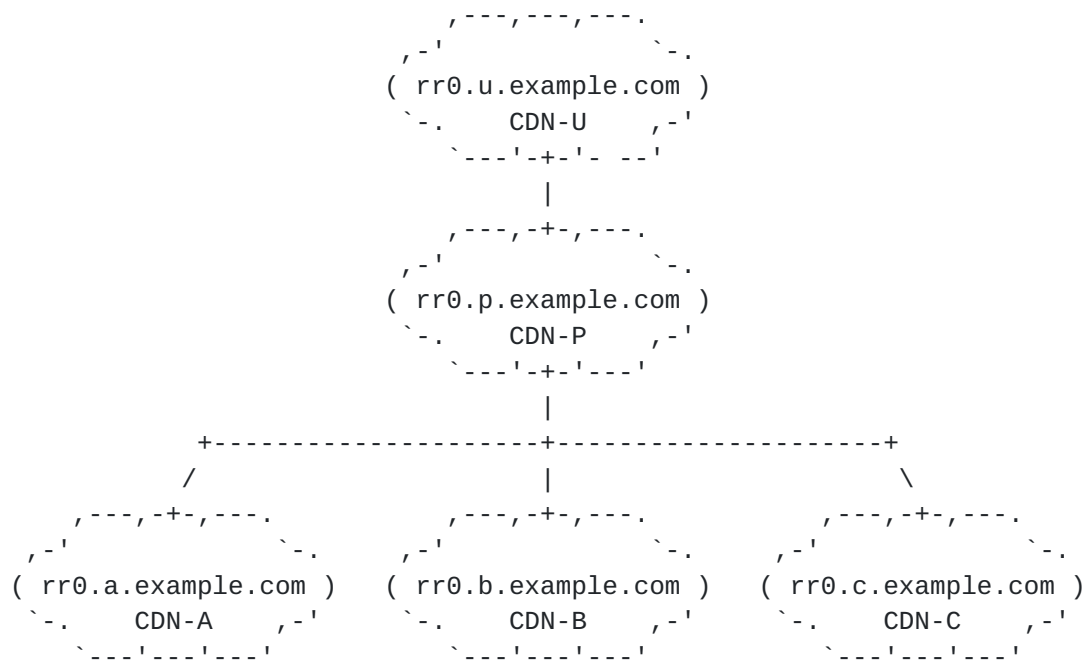


Figure A1: CDNI dCDN Request Router Aggregation

- o None of the four dCDNs (CDNs P, A, B, and C) have global reachability. In this case, each CDN is likely to advertise footprint information with its capabilities, specifying its reachability. When CDN-P advertises capabilities to CDN-U, it may advertise the aggregate footprint of itself and CDNs A, B, and C. Note: CDN-P MAY exclude any dCDN, and consequently its footprint, per its own internal aggregation decision criteria.
- o All four dCDNs (CDNs P, A, B, and C) have global reachability. In this case, none of the CDNs is likely to advertise any footprint information as none have any footprint restrictions. When CDN-P advertises capabilities to CDN-U, the aggregate of all global reachability is global reachability.
- o Some of the four dCDNs (CDNs P, A, B, and C) have global reachability and some do not. In this case, even though some dCDNs do not have global reachability, the aggregate of some dCDNs having global reachability and some not should still be global reachability (for the given capability). When CDN-P advertises capabilities to CDN-U, CDN-P may advertise capabilities for which at least one dCDN has global reach as being supported with global reachability. It is up to the CDN-P request router to properly select a dCDN to process individual client requests and not choose a dCDN whose restricted footprint makes it unsuitable for delivering the requested content.

7.2. Internal Request Router Aggregation

Figure A2 shows CDN-U and CDN-P where CDN-P internally has four request routers: the authoritative request router rr0, and three other request routers rr1, rr2, and rr3. The use of multiple request routers may be used to distribute request routing load across resources, possibly in different geographic regions covered by CDN-P. Similar to Figure A1, the setup shown in Figure A2 requires the authoritative request router rr0 in CDN-P to aggregate capabilities information from downstream request routers rr1, rr2, and rr3. The primary difference between the scenario is that the request routers in Figure A2 are logically within the same CDN-P organization. The same reachability scenarios apply to Figure A2 as with Figure A1.

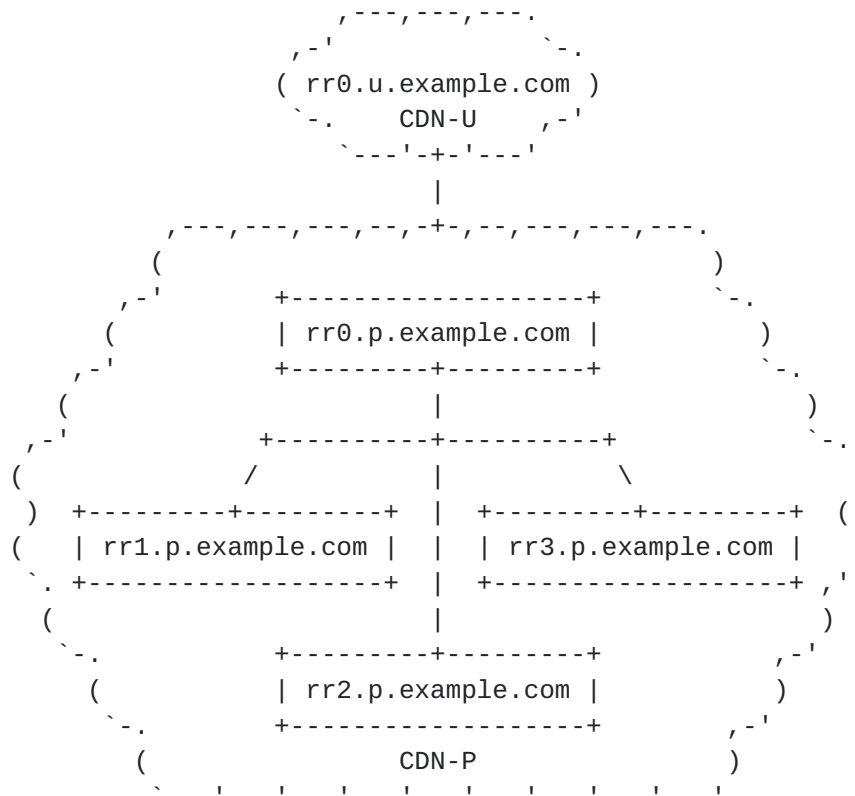


Figure A2: Local CDN Request Router Aggregation

- o None of the four CDN-P request routers have global reachability. In this case, each request router is likely to advertise footprint information with its capabilities, specifying its reachability. When rr0 advertises capabilities to CDN-U, it may advertise the aggregate footprint of itself and rr1, rr2, and rr3.
- o All four CDN-P request routers have global reachability. In this case, none of the request routers is likely to advertise any

footprint information as none has any footprint restrictions. When rr0 advertises capabilities to CDN-U, the aggregate of all global reachability is global reachability.

- o Some of the four CDN-P request routers have global reachability and some do not. In this case, even though some request routers do not have global reachability, the aggregate of some request routers having global reachability and some not should still be global reachability (for the given capability). When rr0 advertises capabilities to CDN-U, CDN-P may advertise capabilities for which at least one request router has global reach as being supported with global reachability. It is up to the authoritative request router rr0 to properly select from the other request routers for any given request, and not choose a request router whose restricted footprint makes it unsuitable for delivering the requested content.

7.3. Internal Capability Aggregation

Figure A3 shows CDN-U and CDN-P where the delivery network of CDN-P is segregated by delivery protocol (e.g., RTSP, HTTP, and RTMP). Figure A3 differs from Figures A1 and A2 in that request router rr0 of CDN-P is not aggregating the capabilities advertisements of multiple other downstream request routers, but rather it is managing the disparate capabilities across resources within its own local CDN. Though not every delivery node has the same protocol capabilities, the aggregate delivery protocol capabilities advertised by CDN-A may include all delivery protocols. Note, Figure A3 should not be construed to imply anything about the coverage areas for each delivery protocol. They may all support the same delivery footprint, or they may have different delivery footprints. It is the responsibility of the request router rr0 to properly assign protocol-appropriate delivery nodes to individual content requests. If certain protocols have limited reachability, CDN-P may advertise footprint restrictions for each protocol.

It should be noted that though the delivery protocol capability was selected for this example, the concept of internal capability aggregation applies to all capabilities as discussed below.

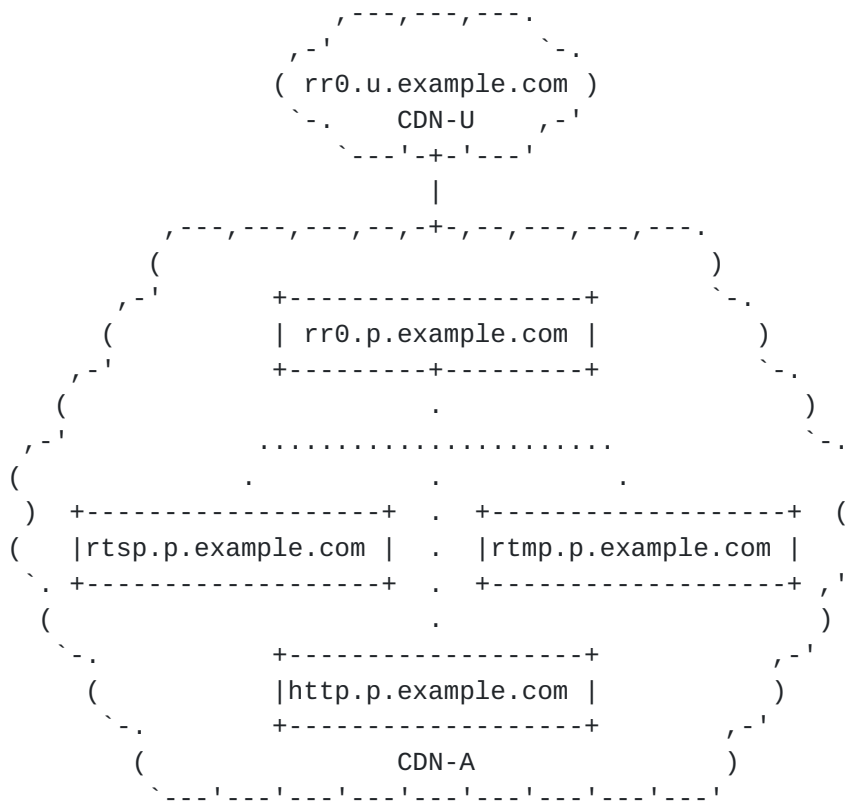


Figure A3: Local CDN Capability Segregation

Another situation in which physical footprint may not matter in an aggregated view has to do with feature support (e.g., new CDNI metadata features or new redirection modes). Situations often arise when phased roll-out of software upgrades, or staging network segregation result in only certain portions of a CDN's resources supporting the new feature set. The dCDN has a few options in this case:

- o Enforce atomic update: The dCDN does not advertise support for the new capability until all resources have been upgraded to support the new capability.
- o Transparent segregation: The dCDN advertises support for the new capability, and when requests are received that require the new capability, the dCDN request router properly selects a resource which supports that capability.
- o Advertised segregation: The dCDN advertises support for the new capability with a footprint restriction allowing the uCDN to make delegation decisions based on the dCDN's limit support.

The level of aggregation employed by the dCDN is likely to vary as

business relationships dictate, however, the capability interface should support all possible modes of operation.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.

8.2. Informative References

- [I-D.ietf-cdni-framework]
Peterson, L., Ed. and B. Davie, Ed., "Framework for CDN Interconnection [draft-ietf-cdni-framework-01](#)", July 2012.
- [I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements [draft-ietf-cdni-requirements-03](#)", June 2012.

Author's Address

Kevin J. Ma
Azuki Systems, Inc.
43 Nagog Park
Acton, MA 01720
USA

Phone: +1 978-844-5100
Email: kevin.ma@azukisystems.com

