Network Working Group M. Caulfield
Internet-Draft K. Leung
Intended status: Standards Track Cisco
Expires: April 26, 2012 October 24, 2011

Content Distribution Network Interconnection (CDNI) Core Metadata draft-caulfield-cdni-metadata-core-00

Abstract

The CDNI Metadata Interface enables interconnected CDNs to exchange content distribution metadata for the purpose of content acquisition and delivery. The CDNI metadata associated with a piece of content provides a downstream CDN with the information necessary for the downstream CDN to service content requests on behalf of an upstream CDN in accordance with the delivery policies defined by the upstream CDN. This document describes the core set of CDNI metadata that all interconnected CDNs must support.

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 April 26, 2012.

Table of Contents

1. Introduction

Several types of metadata flow through content delivery networks. For readability, a number of definitions from [I-D.ietf-cdni-problem-statement] related to metadata are quoted below:

> Content Metadata: This is metadata about Content. Content Metadata comprises:
>
> 1. CDNI Metadata: Content Distribution Metadata with inter-CDN scope. For example, CDNI Metadata may include geo-blocking information (i.e. information defining geographical areas where the content is to be made available or blocked), availability windows (i.e. information defining time windows during which the content is to be made available or blocked) and access control mechanisms to be enforced (e.g. URI signature validation). CDNI Metadata may also include information about desired distribution policy (e.g. prepositioned vs dynamic acquisition) and about where/how a CDN can acquire the content. CDNI Metadata may also include content management information (e.g. request for deletion of Content from Surrogates) across interconnected CDNs. that is relevant to the distribution of the content (and therefore relevant to a CDN involved in the delivery of that content). We refer to this type of metadata as "Content Distribution Metadata". See also the definition of Content Distribution Metadata.
>
> 2. Metadata that is associated with the actual Content (and not directly relevant to the distribution of that Content) or content representation. For example, such metadata may include information pertaining to the Content’s genre, cast, rating, etc as well as information pertaining to the Content representation’s resolution, aspect ratio, etc.
>
> Content Distribution Metadata: The subset of Content Metadata that is relevant to the distribution of the content. This is the metadata required by a CDN in order to enable and control content distribution and delivery by the CDN. In a CDN Interconnection environment, some of the Content Distribution Metadata may have an intra-CDN scope (and therefore need not be communicated between CDNs), while some of the Content Distribution Metadata have an inter-CDN scope (and therefore needs to be communicated between CDNs).
>
> CDNI Metadata: Content Distribution Metadata with inter-CDN scope. For example, CDNI Metadata may include geo-blocking information

(i.e. information defining geographical areas where the content is to be made available or blocked), availability windows (i.e. information defining time windows during which the content is to be made available or blocked) and access control mechanisms to be enforced (e.g. URI signature validation). CDNI Metadata may also include information about desired distribution policy (e.g. prepositioned vs dynamic acquisition) and about where/how a CDN can acquire the content. CDNI Metadata may also include content management information (e.g. request for deletion of Content from Surrogates) across interconnected CDNs.

Interconnecting CDNs necessitates the exchange of the CDNI metadata as defined above. The CDNI metadata associated with a piece of content (or set of contents) provides a downstream CDN with the information necessary for the downstream CDN to service content requests on behalf of an upstream CDN in accordance with the delivery policies defined by the upstream CDN.

The CDNI Metadata Interface is introduced by [I-D.ietf-cdni-problem-statement], and discussed in [I-D.davie-cdni-framework], as one of the four required interfaces for CDN interconnection. The requirements for this interface are specified in [I-D.ietf-cdni-requirements].

This document first describes the core set of CDNI metadata that all interconnected CDNs must support. Then the document describes the relationship between the core metadata and the encoding and transport for exchanging that metadata.

2. Core CDNI Metadata

Although the CDNI Metadata Interface should be flexible enough to support the exchange of arbitrary pieces of metadata, a CDN implementing the interface must support a set of core metadata. The core CDNI metadata represent common policies for content distribution across CDNs. All CDNI metadata may differ on a per-content-item basis or may be shared by a set of content items. The core CDNI Metadata comprises acquisition metadata and delivery metadata.

2.1. Acquisition Metadata

If a downstream CDN receives a request for a content that is not yet cached by that CDN, it will attempt to acquire it from either an upstream CDN or an origin server. The acquisition metadata for a piece of content provides the information needed by a downstream CDN to acquire the missing content. The acquisition metadata includes a prioritized list of content sources. Each content source includes

the following:

1. Protocol - acquisition protocol (e.g. HTTP, FTP, ...).

2. URI - either an explicit URI for acquiring the content or a regex rule for converting from the content request URI to the acquisition URI.

3. Authentication - an object describing the authentication type (e.g. HTTP Basic, Digest, etc.) and any parameters for that type (e.g. username and password).

2.2. Delivery Metadata

Once a piece of content is acquired, the delivery metadata controls where, when, how, and to whom the downstream CDN may deliver that content. The delivery metadata includes a list of permissible delivery profiles. Each profile includes criteria and rules for delivery. Profile criteria include the following:

1. Protocol - delivery protocol (e.g. HTTP, FTP, RTSP).

2. Region - a geographic region identified by country, AS number, or IP subnet.

3. Time window - a time period including a start time and an end time.

If a content request matches all the criteria of a profile, then the rules for that profile should be applied. Profile rules include the following:

1. Allow/deny - flag indicating whether or not the request should be permitted.

2. Authorization - a list of permissible authorization methods and their related parameters (e.g. URL-signing, token-based, etc.).

If a content request matches the criteria of multiple profiles in a list, it should use the rules of the first matching profile.

3. Metadata Encoding

Metadata is encoded as a hierarchy of objects which in practice may be implemented in JavaScript Object Notation (JSON), Extensible Markup Language (XML), or another variant. This section describes the structure of the data but does not prescribe a particular

encoding (such as JSON or XML). The language used below uses generic terms like "lists", "objects", and "fields" to describe the structure of the metadata.

Each piece of content is associated with a CDNIMetadata object which has the following fields:

1. acquisitionOptions - an ordered list of ContentSource objects. The content sources are listed in order of priority with the first being the most desirable option.

2. deliveryProfiles - an ordered list of ContentDelivery objects. Like the content sources, the content delivery profiles are listed in order of priority.

3. metadataScope - a single MetadataScope object.

The CDNIMetadata object fields may be expanded in the future to include a richer set of optional metadata. Proprietary fields may be added with the "x-" prefix.

All fields in the CDNI metadata are optional unless stated otherwise. If a field is missing, its default value should be used. If the value of the field is the default, it need not be included. The default value of a list is the empty list, an object is an empty object, and a string is an empty string unless stated otherwise.

3.1. ContentSource object

The ContentSource object describes a single acquisition point that a downstream CDN may contact to acquire the content. This object has the following fields:

1. protocol - a string containing the name of the protocol that may be used to acquire the content (e.g. HTTP, FTP, ...). The default protocol is "http".

2. uriType - a string containing either "explicit" or "regex" which dictates how the uri field should be interpreted. If the type is "explicit", the URI is to be used as is. If the type is "regex" then the uri field specifies a regex substitution that should be performed on the content URL for mapping to the explicit URI. The default uriType is "explicit".

3. uri - a string containing either the URI of the content source or a regex substitution to generate the acquisition URI, depending on the value of uriType. This field is required in a ContentSource object and its value may not be empty.

4. auth - a single Authentication object. If the auth field is missing, then authentication is not required for this ContentSource.

3.2. ContentDelivery object

The ContentDelivery object describes a permissible delivery profile for the content. This object has the following fields:

1. deliveryCriteria - an ordered list of DeliveryCriteria objects.

2. deliveryRules - a single DeliveryRules object.

3.3. MetadataScope object

The MetadataScope object indicates that a CDNIMetadata object applies to more than one content and may be used as an optimization by downstream CDNs to avoid refetching duplicate metadata. If a new content request meets the criteria in this object, then the entire CDNIMetadata object applies to that request and the downstream CDN need not refetch it. The object includes the following fields:

1. host - an optional string containing a regex that could be checked against the Host header of an incoming HTTP request.

2. resource - an optional string containing a regex that could be checked against the requested resource name in an incoming HTTP request, to determine if the metadata object is associated to the requested content item.

3. protocol - an optional string containing a protocol name that may be checked against the client request protocol (e.g. "http" or "ftp").

If the MetadataGroup object is missing from the CDNIMetadata object, then the CDNIMetadata object only applies to the requested content and may not be reused for different content requests.

3.4. Authentication object

The Authentication object describes an authentication type and its parameters. It provides information for content acquisition such that the downstream CDN can be authenticated as a client when acquiring content from an upstream CDN or an origin server. The Authentication object contains the following fields:

1. type - a string containing the authentication type "basic" or "digest". The type dictates which optional fields are present

and valid in the rest of the object. The "basic" and "digest" types refer to HTTP Basic and Digest access authentication [RFC2617] respectively.

2. username - a string containing the username for "basic" and "digest" types.

3. password - a string containing the password for "basic" and "digest" types.

3.5. DeliveryCriteria object

The DeliveryCriteria object specifies a set of criteria to match against incoming content requests including protocol, region, and time window. The object includes the following fields:

1. protocol - a string containing the name of a protocol to match (e.g. "http", "ftp", ...).

2. region - a single Region object.

3. timeWindow - a single TimeWindow object.

3.6. DeliveryRules object

The DeliveryRules object describes the rules to apply to a particular content request in order to deliver a piece of content to the user agent on behalf of an upstream CDN. This object includes the following fields:

1. allow - a boolean stating whether or not delivery is permitted.

2. auth - a single Authorization object.

3.7. Region object

The Region object specifies a region where the content is either allowed or disallowed. A region may be described in three ways, only one of which needs to be present per Region object. The object includes the following fields:

1. country - a string containing the country code for the region.

2. bgpAs - a number containing the BGP AS identifier of the region.

3. subnet - a string containing a dotted decimal IP address and subnet mask.

3.8. TimeWindow object

The TimeWindow object specifies a time period when a content is available or not available. The object includes the following fields:

1. startTime - a string containing an ISO 8601 formatted date and time in UTC.

2. endTime - a string containing the same format as startTime.

3.9. Authorization object

The Authorization object describes an authorization type and its parameters. It provides information for content delivery such that the user agent can be authenticated as a client when requesting content from a downstream CDN. The Authorization object contains the following fields:

1. type - a string containing the authorization type "url-signing" or "url-token". The type dictates which optional fields are present and valid in the rest of the object. The "url-signing" type refers to URL signing authorization. The "url-token" type refers to token-based authorization.

2. algo - a string containing the signature algorithm (e.g. "md5", "sha-1", etc.).

3. symmetric - a boolean if true, URL signing uses symmetric keys, otherwise asymmetric.

4. key - a number containing the public key for verifying signatures, only valid if "symmetric" field is set to false.

[Editor’s note: parameters for URL signing and URL token authorization schemes are TBD. Private key provisioning and distribution are outside the scope of the CDNI Metadata Interface.]

3.10. Reference object

In order to avoid refetching large or common objects, any object may be replaced by a Reference object. For example, the ContentSource object could be replaced by a Reference object which points to a URI. The downstream CDN should then request the referenced object in order to complete the metadata. This object includes the following fields:

1. ref - a string containing a URI which points to the actual object.

A Reference object may be distinguished from any other type of object by checking for the presence of the "ref" field with a non-empty value.

4. Metadata Transport

Metadata objects (JSON, XML, etc.) are assumed to be transported over HTTP. This section describes the relationship between the encoding and transport layers.

Given a content request from an end client, when the downstream CDN needs to acquire the metadata associated with that content, the downstream CDN uses an HTTP GET to query the upstream CDN metadata server for the metadata object.

The parameters to the query request are identical to the fields of the MetadataScope object discussed earlier. This similarity is intentional and allows the downstream CDN to avoid refetching the same metadata for two different pieces of content (if the metadata is the same). The query parameters are extracted from an incoming content request and are as follows:

1. host - the value of the Host header in an HTTP request.

2. resource - the resource on the request line of the HTTP request.

3. protocol - the protocol of the incoming request.

After extracting these fields from the content request, the downstream CDN should first check its existing cache of metadata objects for possible matches (using the MetadataScope object). If no match is found, the downstream CDN should then form a request to the upstream CDN metadata server, appending the host, resource, and protocol as query string parameters.

The metadata server will respond with a CDNIMetadata object encoded in JSON, XML, or some other variant.

5. CDNI Metadata Interface Bootstrapping

This document makes a number of assumptions regarding the information available to the downstream CDN which is not part of the CDNI Core Metadata. Information such as how a downstream CDN learns the address or hostname of the upstream metadata server is briefly described below.

In the simplest case, the Control Interface will provision the URI of the metadata server to the downstream CDN and all metadata requests will be sent directly to this server. The Control Interface could also provision an alternative URI in case the primary server is unreachable.

In the case of multiple potential upstream CDNs, the downstream CDN must decide which metadata server should handle its request. It is expected that the downstream CDN will be able to determine the upstream CDN that redirected that particular request from information contained in the received request (e.g. via the URI in case of HTTP redirection across CDNs). With knowledge of which upstream CDN routed the request, the downstream CDN can choose the correct metadata server.

6. Compliance with CDNI Requirements

This section reviews compliance of the solution proposed in this document against the relevant set of requirements from [I-D.ietf-cdni-requirements].

[Editor's note: the compliance information will be provided in subsequent versions]

7. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Security Considerations

The CDNI Metadata Interface is expected to be secured as a function of the transport protocol (e.g. HTTP authentication).

If a malicious metadata server is contacted by a downstream CDN, the malicious server may provide metadata to the downstream CDN which denies service for any piece of content to any user agent. The malicious server may also provide metadata which directs a downstream CDN to a malicious origin server instead of the actual origin server.

9. Acknowledgements

The authors would like to thank Francois le Faucheur for his input and review.

10. Normative References

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-00 (work in progress), September 2011.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-01 (work in progress), October 2011.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

Appendix A. Example Metadata

Below is an example of a JSON object encoding a piece of CDNI metadata. This metadata object applies to any content request with a hostname of "origin.example.com" and a resource identifier matching the regular expression "*/movies/*". The "metadataScope" field summarizes these properties. There is a single acquisition option as seen in the "acquisitionOptions" list. The "deliveryProfiles" information restricts users in subnet "10.1.2.0/24" or using RTSP from accessing this content and allows users using HTTP.

```
{
  "acquisitionOptions" :
  [
    {
      "protocol" : "http",
      "uri" : "http://origin.example.com/movies/content.mpg",
      "auth" :
```

```
        {
          "type" : "basic",
          "username" : "abcd",
          "password" : "pass123"
        }
      }
    ],
    "deliveryProfiles" :
    [
      {
        "deliveryCriteria" :
        [
          {
            "region" : { "subnet" : "10.1.2.0/24" },
          },
          {
            "protocol" : "rtsp"
          }
        ],
        "deliveryRules" :
        {
          "allow" : false
        }
      },
      {
        "deliveryCriteria" :
        [
          {
            "protocol" : "http"
          }
        ],
        "deliveryRules" :
        {
          "allow" : true
        }
      }
    ],
    "metadataScope" :
    {
      "host" : "origin.example.com",
      "resource" : "*/movies/*"
    }
  }
```

Authors' Addresses

Matt Caulfield
Cisco Systems
1414 Massachusetts Ave
Boxborough, MA 01719
USA

Phone: +1 978 936 9307
Email: mcaulfie@cisco.com

Kent Leung
Cisco Systems
3625 Cisco Way
San Jose 95134
USA

Phone: +1 408 526 5030
Email: kleung@cisco.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

B. Davie, Ed.
Cisco Systems, Inc.
L. Peterson, Ed.
Verivue, Inc.
October 31, 2011

Framework for CDN Interconnection
draft-davie-cdni-framework-01

Abstract

This document presents a framework for Content Distribution Network Interconnection (CDNI). The purpose of the framework is to provide an overall picture of the problem space of CDNI and to describe the relationships among the various components necessary to interconnect CDNs. CDN Interconnection requires the specification of several interfaces and mechanisms to address issues such as request routing, metadata exchange, and the acquisition of content by one CDN from another. The intent of this document is to outline what each interface needs to accomplish, and to describe how these interfaces and mechanisms fit together, while leaving their detailed specification to other documents.

Table of Contents

1. Introduction

The interconnection of Content Distribution Networks (CDNs) is motivated by several use cases, such as those described in [I-D.ietf-cdni-use-cases]. The overall problem space for CDN Interconnection is described in [I-D.ietf-cdni-problem-statement]. The purpose of this document is to provide an overview of the various components necessary to interconnect CDNs. CDN Interconnection requires the specification of several interfaces and mechanisms to address issues such as request routing, metadata exchange, and the acquisition of content by one CDN from another. The intent of this document is to describe how these interfaces and mechanisms fit together, leaving their detailed specification to other documents. We make extensive use of message flow examples to illustrate the operation of interconnected CDNs, but these examples should be considered illustrative rather than prescriptive.

1.1. Terminology

This document draws freely on the terminology defined in [RFC3466] and [I-D.ietf-cdni-problem-statement].

We also introduce the following terms:

CDN Domain: a host name (FQDN) at the beginning of a URL, representing a set of content that is served by a given CDN. For example, in the URL http://cdn.csp.com/...rest of url..., the CDN domain is cdn.csp.com.

Distinguished CDN Domain: a CDN domain that is allocated by a CDN for the purposes of communication with a peer CDN, but which is not found in client requests. Such CDN domains may be used for inter-CDN acquisition, or as redirection targets, and enable a CDN to distinguish a request from a peer CDN from an end-user request.

Recursive CDNI request routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can query the Downstream CDN Request Routing system via the CDNI Request Routing interface (or use information cached from earlier similar queries) to find out how the Downstream CDN wants the request to be redirected, which allows the Upstream CDN to factor in the Downstream CDN response when redirecting the user agent. This approach is referred to as "recursive" CDNI request routing. Note that the Downstream CDN may elect to have the request redirected directly to a Surrogate inside the Downstream CDN, to the Request-Routing System of the Downstream CDN, to another CDN, or to any other system that the Downstream CDN sees as fit for handling the redirected request.

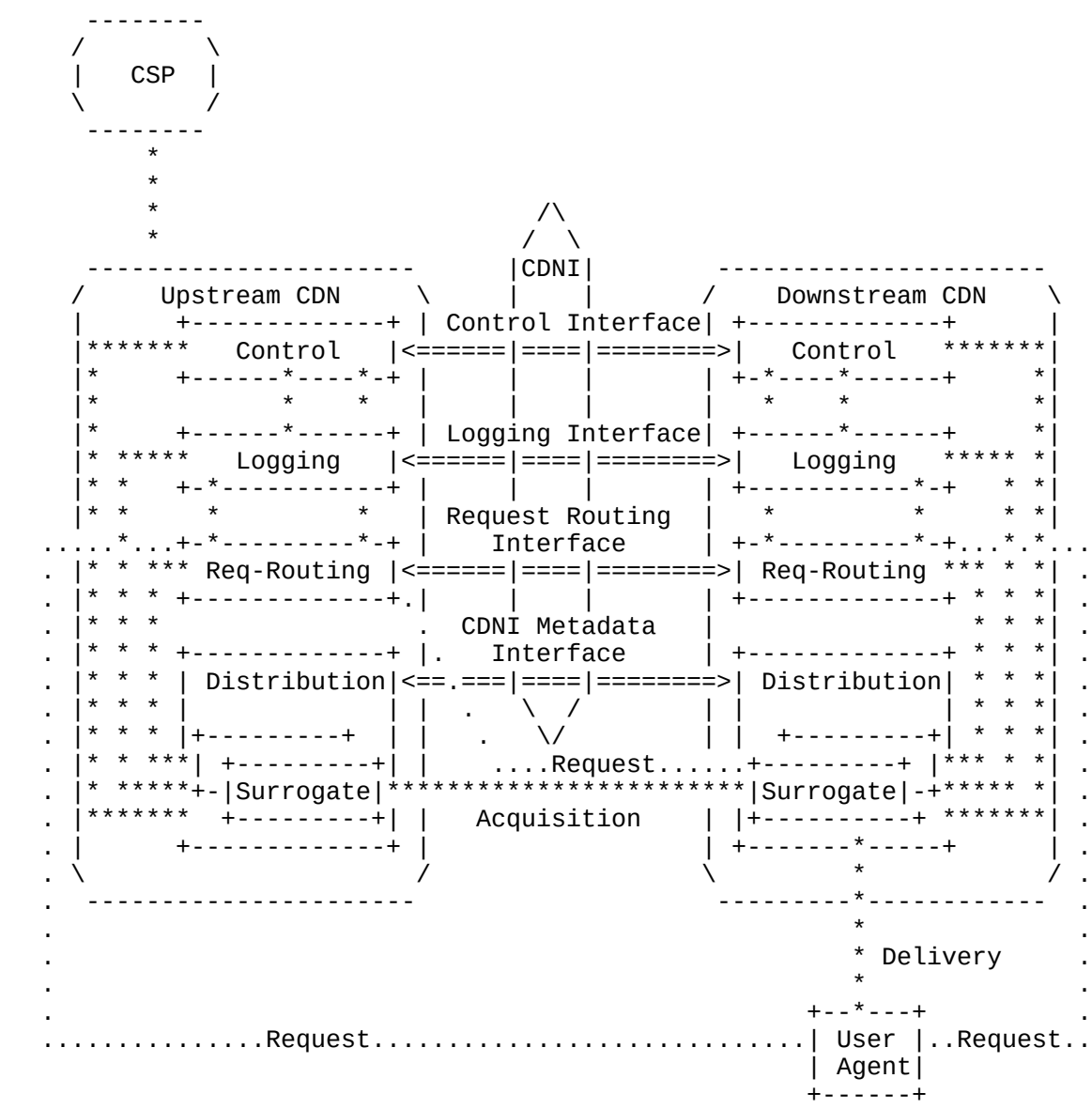
Iterative CDNI Request Routing: When an Upstream CDN elects to redirect a request towards a Downstream CDN, the Upstream CDN can base its redirection purely on a local decision (and without attempting to take into account how the Downstream CDN may in turn redirect the user agent). In that case, the Upstream CDN redirects the request to the request routing system in the Downstream CDN, which in turn will decide how to redirect that request: this approach is referred to as "iterative" CDNI request routing.

Synchronous CDNI operations: operations between CDNs that happen during the process of servicing a user request, i.e. between the time that the user agent begins its attempt to obtain content and the time at which that request is served.

Asynchronous CDNI operations: operations between CDNs that happen independently of any given user request, such as advertisement of footprint information or pre-positioning of content for later delivery.

1.2. Reference Model

This document uses the reference model in Figure 1 as originally created in [I-D.ietf-cdni-problem-statement].



<==> interfaces inside the scope of CDNI

**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Model and CDNI Interfaces

We note that while some interfaces in the reference model are "out of scope" for the CDNI WG (in the sense that there is no need to define new protocols for those interfaces) we still need to refer to them in this document to explain the overall operation of CDNI.

We also note that, while we generally show only one uCDN serving a given CSP, it is entirely possible that multiple uCDNs can serve a single CSP. In fact, this situation effectively exists today in the sense that a single CSP can connect to more than one CDN today.

Definitions of the four CDNI interfaces follow. More discussion of these interfaces appears in Section 4.

- Control Interface: Operations to discover, initialize, and parameterize the other CDNI interfaces. Once established, all runtime control over CDNI behavior is under the purview of one of these other interfaces.

- Request Routing Interface: Operations to determine what CDN (and optionally what surrogate within a CDN) is to serve end-user's requests. May include a combination of:

  * Asynchronous operations to exchange routing information (e.g., the network footprint served by a given CDN) that enables CDN selection for subsequent user requests; and

  * Synchronous operations to select a delivery CDN (surrogate) for a given user request.

- Metadata Interface: Operations to communicate metadata that governs the how content is delivered by interconnected CDNs. Examples of CDNI metadata include geo-blocking directives, availability windows, access control mechanisms, and purge directives. May include a combination of:

  * Asynchronous operations to exchange metadata that govern subsequent user requests for content; and

  * Synchronous operations that govern behavior for a given user request for content.

- Logging Interface: Operations that allow interconnected CDNs to exchange relevant activity logs. May include a combination of:

  * Real-time exchanges, suitable for runtime traffic monitoring; and

* Off-line exchanges, suitable for analytics and billing.

1.3. Structure Of This Document

The remainder of this document is organized as follows:

- Section 2 describes some essential building blocks for CDNI, notably the various options for redirecting user requests to a given CDN.
- Section 3 provides a number of illustrative examples of various CDNI operations.
- Section 4 describes the functionality of the four main CDNI interfaces.
- Section 5 shows how various deployment models of CDNI may be achieved using the defined interfaces.
- Section 6 describes the trust model of CDNI and the issues of transitive trust in particular that CDNI raises.

2. Building Blocks

2.1. Request Redirection

At its core, CDN Interconnection requires the redirection of requests from one CDN to another. For any given request that is received by an upstream CDN, it will either respond to the request directly, or somehow redirect the request to a downstream CDN. Two main mechanisms are available for redirecting a request to a downstream CDN. The first leverages the DNS name resolution process and the second uses in-protocol redirection mechanisms such as the HTTP 302 redirection response. We discuss these below as background before discussing some examples of their use in Section 3.

2.1.1. DNS Redirection

DNS redirection is based on returning different IP addresses for the same DNS name, for example, to balance server load or to account for the client’s location in the network. A DNS server, sometimes called the Local DNS (LDNS), resolves DNS names on behalf of an end-user. The LDNS server in turn queries other DNS servers until it reaches the authoritative DNS server for the CDN-domain. The network operator typically provides the LDNS server, although the user is free to choose other DNS servers (e.g., OpenDNS, Google Public DNS).

The advantage of DNS redirection is that it is completely transparent to the end user--the user sends a DNS name to the LDNS server and gets back an IP address. On the other hand, DNS redirection is problematic because the DNS request comes from the LDNS server, not the end-user. This may affect the accuracy of server selection that is based on the user's location. The transparency of DNS redirection is also a problem in that there is no opportunity to modify the path component of the URL being accessed by the client. We consider two main forms of DNS redirection: simple and CNAME-based.

In simple DNS redirection, the authoritative DNS server for the name simply returns an IP address from a set of possible IP addresses. The answer is chosen from the set based on characteristics of the set (e.g., the relative loads on the servers) or characteristics of the client (e.g., the location of the client relative to the servers). Simple redirection is straightforward. The only caveats are (1) there is a limit to the number of delivery nodes a single DNS server can manage; and (2) DNS responses are cached by downstream servers so the TTL on the response must be set to an appropriate value so as to preserve the timeliness of the redirection.

In CNAME-based DNS redirection, the authoritative server returns a CNAME response to the DNS request, telling the LDNS server to restart the name lookup using a new name. A CNAME is essentially a symbolic link in the DNS namespace, and like a symbolic link, redirection is transparent to the client--the LDNS server gets the CNAME response and re-executes the lookup. Only when the name has been resolved to an IP address does it return the result to the user. Note that DNAME would be preferable to CNAME if it becomes widely supported.

2.1.2. HTTP Redirection

HTTP redirection makes use of the "302" redirection response of the HTTP protocol. This response contains a new URL that the application should fetch instead of the original URL. By changing the URL appropriately, the server can cause the user to redirect to a different server. The advantages of 302 redirection are that (1) the server can change the URL fetched by the client to include, for example, both the DNS name of the particular server to use, as well as the original HTTP server that was being accessed; and (2) the client sends the HTTP request to the server, so that its IP address is known and can be used in selecting the server.

The disadvantages of HTTP redirection are (1) it is visible to the application, so it requires application support and may affect the application behavior (e.g., web browsers will not send cookies if the URL changes to a different domain); (2) HTTP is a heavy-weight protocol layered on TCP so it has relatively high overhead; and (3)

the results of HTTP redirection are not cached so that all redirections must go through to the server.

3. Overview of CDNI Operation

To provide a big-picture overview of the various components of CDN Interconnection, we walk through a "day in the life" of a content item that is made available via a pair of interconnected CDNs. This will serve to illustrate many of the functions that need to be supported in a complete CDNI solution. We give examples using both DNS-based and HTTP-based redirection. We begin with very simple examples and then how additional capabilities, such as recursive request redirection and content removal, might be added.

Before walking through some specific examples, we present a high-level view of the operations that may take place. This high-level overview is illustrated in Figure 2. Note that most operations will involve only a subset of all the messages shown below, and that the order and number of operations may vary considerably, as more detailed examples illustrate below.

The following shows Operator A as the upstream CDN (uCDN) and Operator B as the downstream CDN (dCDN), where the former has a relationship with a content provider and the latter being the best CDN to deliver content to the end-user. The interconnection relationship may be symmetric between these two CDN operators, but for simplicity we show the interaction in one direction only.

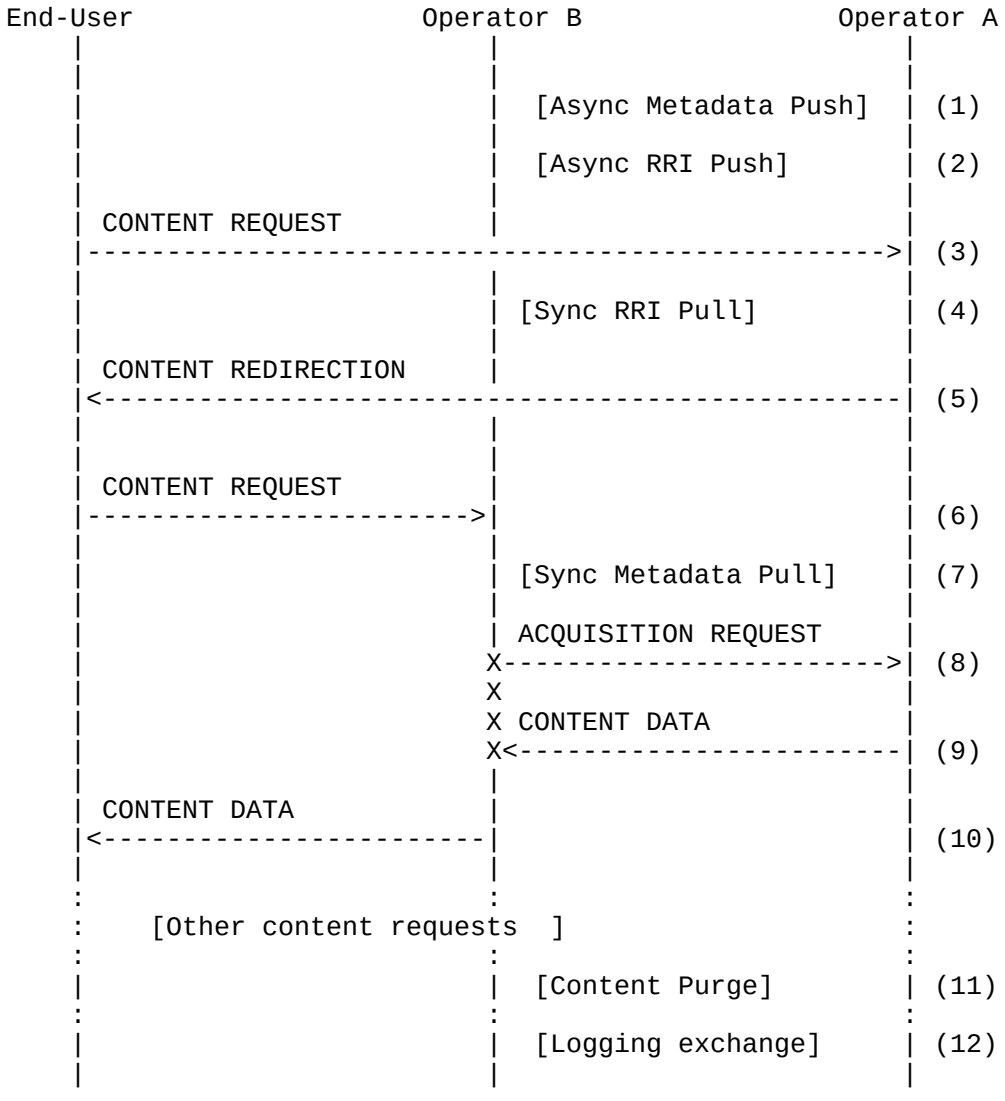


Figure 2: Overview of Operation

The operations shown in the Figure are as follows:

1. Prior to any content request, metadata may be asynchronously pushed from uCDN to dCDN so that it is available in readiness for later content requests.

2. dCDN may advertise information relevant to its delivery capabilities (e.g. geographic footprint, reachable address prefixes) prior to any content requests being redirected.

3. A content request from a user agent arrives at uCDN.

4. uCDN may synchronously request information from dCDN regarding its delivery capabilities to decide if dCDN is a suitable target for redirection of this request.

5. uCDN redirects the request to dCDN by sending some response (DNS, HTTP) to the user agent.

6. The user agent requests the content from dCDN.

7. dCDN may synchronously request metadata related to this content from uCDN, e.g. to decide whether to serve it.

8. If the content is not already in a suitable cache in dCDN, dCDN may acquire it from uCDN.

9. The content is delivered to dCDN from uCDN.

10. The content is delivered to the user agent by dCDN.

11. Some time later, perhaps at the request of the CSP (not shown) uCDN may instruct dCDN to purge the content to ensure it is not delivered again.

12. After one or more content delivery actions by dCDN, a log of delivery actions may be provided to uCDN.

The following sections show some more specific examples of how these operations may be combined to perform various delivery, control and logging operations across a pair of CDNs.

3.1. Preliminaries

Initially, we assume that there is at least one CSP that has contracted with an upstream CDN (uCDN) to deliver content on its behalf. We are not particularly concerned with the interface between the CSP and uCDN, other than to note that it is expected to be the same as in the "traditional" (non-interconnected) CDN case. Existing mechanisms such as DNS CNAMEs or HTTP redirects (Section 2) can be used to direct a user request for a piece of content from the CSP towards the CSP’s chosen upstream CDN.

We use the term "CDN-domain" to refer to the host name (a FQDN) at

the beginning of each URL. We assume Operator A provides an upstream CDN that serves content on behalf of a CSP with CDN-domain cdn.csp.com. We assume that Operator B provides a downstream CDN. An end user at some point makes a request for URL

http://cdn.csp.com/...rest of url...

It may well be the case that cdn.csp.com is just a CNAME for some other CDN-domain (such as csp.op-a.net). Nevertheless, the HTTP request in the examples that follow is assumed to be for the example URL above.

Our goal is to enable content identified by the above URL to be served by the CDN of operator B. In the following sections we will walk through some scenarios in which content is served, as well as other CDNI operations such as the removal of content from a downstream CDN.

3.2. HTTP Redirect Example

In this section we walk through a simple, illustrative example using HTTP redirection from uCDN to dCDN. The example also assumes the use of HTTP redirection inside uCDN and dCDN; however, this is independent of the choice of redirection approach across CDNs, so an alternative example could be constructed still showing HTTP redirection from uCDN to dCDN but using DNS for handling of request inside each CDN.

We assume for this example that Operators A and B have established an agreement to interconnect their CDNs, with A being upstream and B being downstream. (It is likely that the agreement would be made in both directions, but we focus on just one here for clarity.)

The operators agree that a CDN-domain peer-a.op-b.net will be used as the target of redirections from uCDN to dCDN. The name of this domain must be communicated by some means to each CDN. (This could be established out-of-band or via a CDNI interface.) We refer to this domain as a "distinguished" CDN domain to convey the fact that its use is limited to the interconnection mechanism; such a domain is never embedded in URLs that end-users request.

The operators must also agree on some distinguished CDN-domain that will be used for inter-CDN acquisition of CSP’s content from uCDN by dCDN. In this example, we’ll use op-b-acq.op-a.net.

The operators must also exchange information regarding which requests dCDN is prepared to serve. For example, dCDN may be prepared to serve requests from clients in a given geographical region or a set

of IP address prefixes. This information may again be provided out of band or via a defined interface.

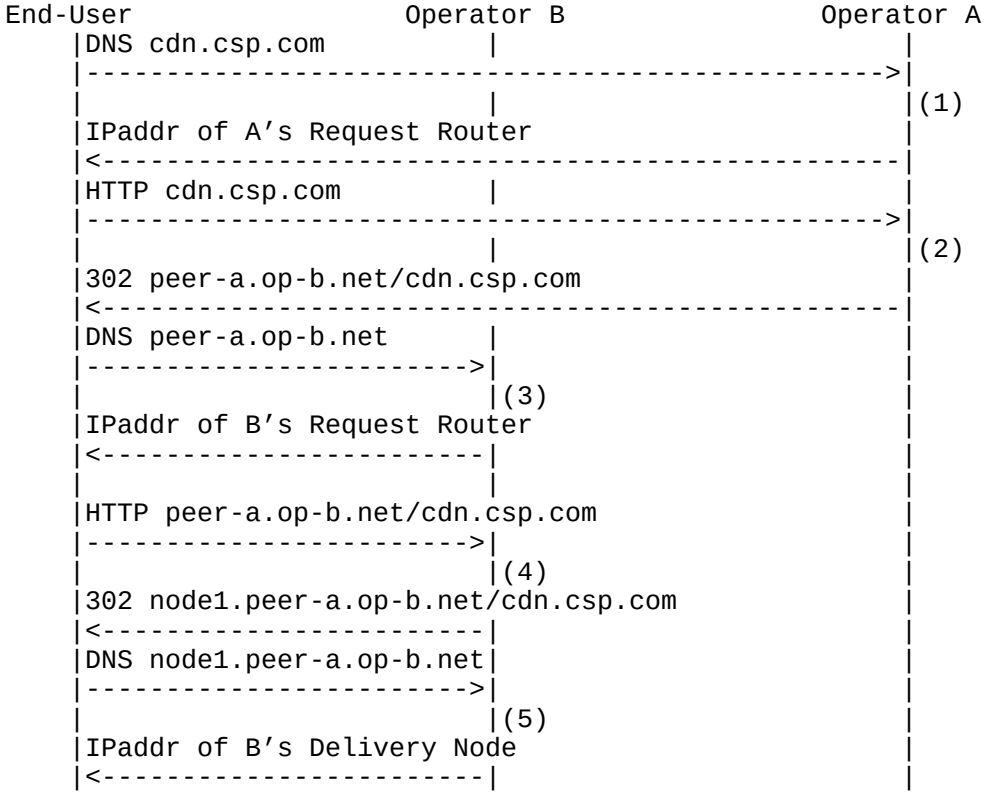
DNS must be configured in the following way:

- The content provider must be configured to make operator A the authoritative DNS server for cdn.csp.com (or to return a CNAME for cdn.csp.com for which operator A is the authoritative DNS server).
- Operator A must be configured so that a DNS request for op-b-acq.op-a.net returns a request router in Operator A.
- Operator B must be configured so that a DNS request for peer-a.op-b.net/cdn.csp.com returns a request router in Operator B.

Figure 3 illustrates how a client request for

http://cdn.csp.com/...rest of url...

is handled.



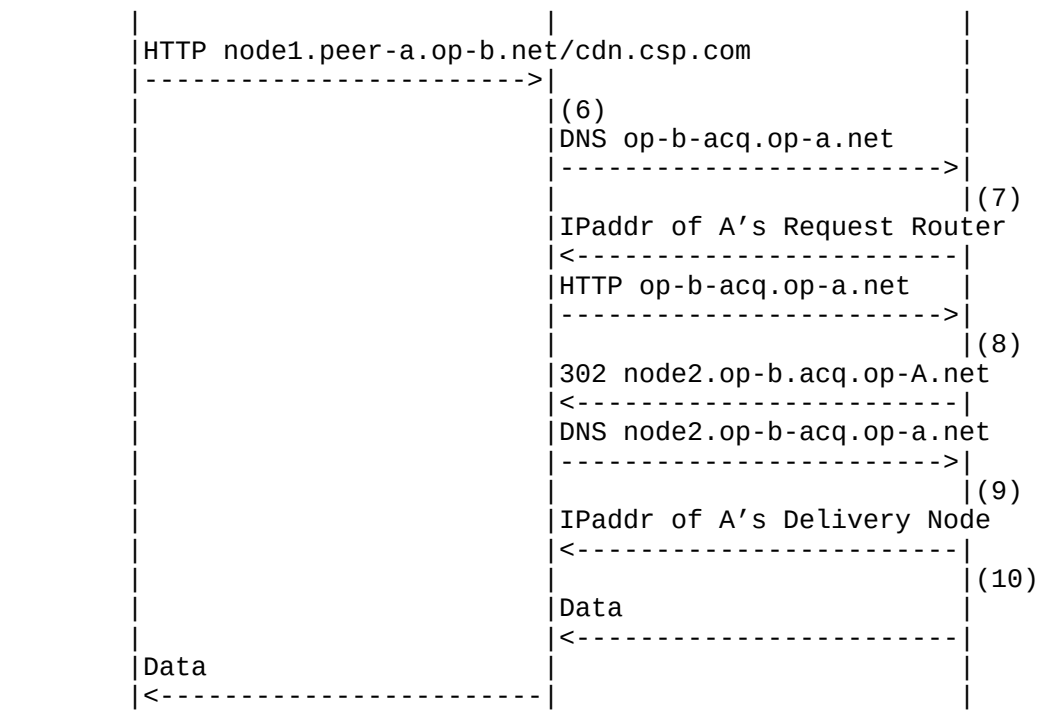


Figure 3: Request Trace for HTTP redirection method

The steps illustrated in the figure are as follows:

1. A DNS resolver for Operator A processes the DNS request for its customer based on CDN-domain cdn.csp.com. It returns the IP address of a request router in Operator A.

2. A Request Router for Operator A processes the HTTP request and recognizes that the end-user is best served by another CDN--specifically one provided by Operator B--and so it returns a 302 redirect message for a new URL constructed by "stacking" Operator B's distinguished CDN-domain (peer-a.op-b.net) on the front of the original URL. (Note that more complex URL manipulations are possible, such as replacing the initial CDN-domain by some opaque handle.)

3. The end-user does a DNS lookup using Operator B's distinguished CDN-domain (peer-a.op-b.net). B's DNS resolver returns the IP address of a request router for Operator B. Note that if request routing within dCDN was performed using DNS instead of HTTP redirection, B's DNS resolver would also behave as the request router and directly return the IP address of a delivery node.

4. The request router for Operator B processes the HTTP request and selects a suitable delivery node to serve the end-user request, and returns a 302 redirect message for a new URL constructed by replacing the hostname by a subdomain of the Operator B's distinguished CDN-domain that points to the selected delivery node.

5. The end-user does a DNS lookup using Operator B's delivery node subdomain (node1.peer-a.op-b.net). B's DNS resolver returns the IP address of the delivery node.

6. The end-user requests the content from B's delivery node. In the case of a cache hit, steps 6, 7, 8, 9 and 10 below do not happen, and the content data is directly returned by the delivery node to the end-user. In the case of a cache miss, the content needs to be acquired by dCDN from uCDN (not the CSP). The distinguished CDN-domain peer-a.op-b.net indicates to dCDN that this content is to be acquired from uCDN; stripping the CDN-domain reveals the original CDN-domain cdn.csp.com and dCDN may verify that this CDN-domain belongs to a known peer (so as to avoid being tricked into serving as an open proxy). It then does a DNS request for an inter-CDN acquisition CDN-domain as agreed above (in this case, op-b-acq.op-a.net).

7. Operator A's DNS resolver processes the DNS request and returns the IP address of a request router in operator A.

8. The request router for Operator A processes the HTTP request from Operator B delivery node. Operator A request router recognizes that the request is from a peer CDN rather than an end-user because of the dedicated inter-CDN acquisition domain (op-b-acq.op-a.net). (Note that without this specially defined inter-CDN acquisition domain, operator A would be at risk of redirecting the request back to operator B, resulting in an infinite loop). The request router for Operator A selects a suitable delivery node in uCDN to serve the inter-CDN acquisition request and returns a 302 redirect message for a new URL constructed by replacing the hostname by a subdomain of the Operator A's distinguished inter-CDN acquisition domain that points to the selected delivery node.

9. Operator A DNS resolver processes the DNS request and returns the IP address of the delivery node in operator A.

10. Operator A serves content for the requested CDN-domain to dCDN. Although not shown, it is at this point that Operator A processes the rest of the URL: it extracts information identifying the origin server, validates that this server has

been registered, and determines the content provider that owns the origin server. It may also perform its own content acquisition steps if needed before returning the content to dCDN.

3.2.1. Comments on the example

The main advantage of this design is that it is simple: each CDN need only know the distinguished CDN-domain for each peer, with the upstream CDN "pushing" the downstream CDN-domain onto the URL as part of its redirect (step 2) and the downstream CDN "popping" its CDN-domain off the URL to expose a CDN-domain that the upstream CDN can correctly process. Neither CDN needs to be aware of the internal structure of the other's URLs. Moreover, the inter-CDN redirection is entirely supported by a single HTTP redirect; neither CDN needs to be aware of the other's internal redirection mechanism (i.e., whether it is DNS or HTTP based).

One disadvantage is that the end-user's browser is redirected to a new URL that is not in the same domain of the original URL. This has implications on a number of security or validation mechanisms sometimes used on endpoints. For example, it is important that any redirected URL be in the same domain (e.g., csp.com) if the browser is expected to send any cookies associated with that domain. As another example, some video players enforce validation of a cross domain policy that needs to allow for the domains involved in the CDN redirection. These problems are generally soluble, but the solutions complicate the example, so we do not discuss them further in this version of the draft.

We note that this example begins to illustrate some of the interfaces that may be required for CDNI, but does not require all of them. For example, obtaining information from dCDN regarding the set of client IP addresses or geographic regions it might be able to serve is an aspect of the request routing interface. Important configuration information such as the distinguished names used for redirection and inter-CDN acquisition could also be conveyed via a CDNI interface (e.g., perhaps the control interface). The example also shows how existing HTTP-based methods suffice for the acquisition interface. Arguably, the absolute minimum metadata required for CDNI is the information required to acquire the content, and this information was provided "in-band" in this example by means of the URI handed to the client in the HTTP 302 response. Hence, there is no explicit metadata interface invoked in this example. There is also no explicit logging interface discussed in this example.

We also note that the step of deciding when a request should be redirected to dCDN rather than served by uCDN has been somewhat

glossed over. It may be as simple as checking the client IP address against a list of prefixes, or it may be considerably more complex, involving a wide range of factors, such as the geographic location of the client (perhaps determined from a third party service), CDN load, or specific business rules.

This example uses the "iterative" CDNI request routing approach. That is, uCDN performs part of the request routing function to determine that dCDN should serve the request, and then redirects the client to a request router in dCDN to perform the rest of the request routing function. If request routing is performed in the dCDN using HTTP redirection, this translates in the end-user experiencing two successive HTTP redirections. By contrast, the alternative approach of "recursive" CDNI request routing effectively coalesces these two successive HTTP redirections into a single one, sending the end-user directly to the right delivery node in the dCDN. This "recursive" CDNI request routing approach is discussed in the next section.

3.3. Recursive Redirection Example

The following example builds on the previous one to illustrate the use of the Request Routing interface to enable "recursive" CDNI request routing. We build on the HTTP-based redirection approach because it illustrates the principles and benefits clearly, but it is equally possible to perform recursive redirection when DNS-based redirection is employed.

In contrast to the prior example, the operators need not agree in advance on a CDN-domain to serve as the target of redirections from uCDN to dCDN. The operators still must agree on some distinguished CDN-domain that will be used for inter-CDN acquisition of CSP's content by dCDN. In this example, we'll use op-b-acq.op-a.net.

The operators must also exchange information regarding which requests dCDN is prepared to serve. For example, dCDN may be prepared to serve requests from clients in a given geographical region or a set of IP address prefixes. This information may again be provided out of band or via a defined protocol.

DNS must be configured in the following way:

- The content provider must be configured to make operator A the authoritative DNS server for cdn.csp.com (or to return a CNAME for cdn.csp.com for which operator A is the authoritative DNS server).

- Operator A must be configured so that a DNS request for op-b-acq.op-a.net returns a request router in Operator A.

- Operator B must be configured so that a request for node1.op-b.net/cdn.csp.com returns the IP address of a delivery node. Note that there might be a number of such delivery nodes.

Figure 3 illustrates how a client request for

http://cdn.csp.com/...rest of url...

is handled.

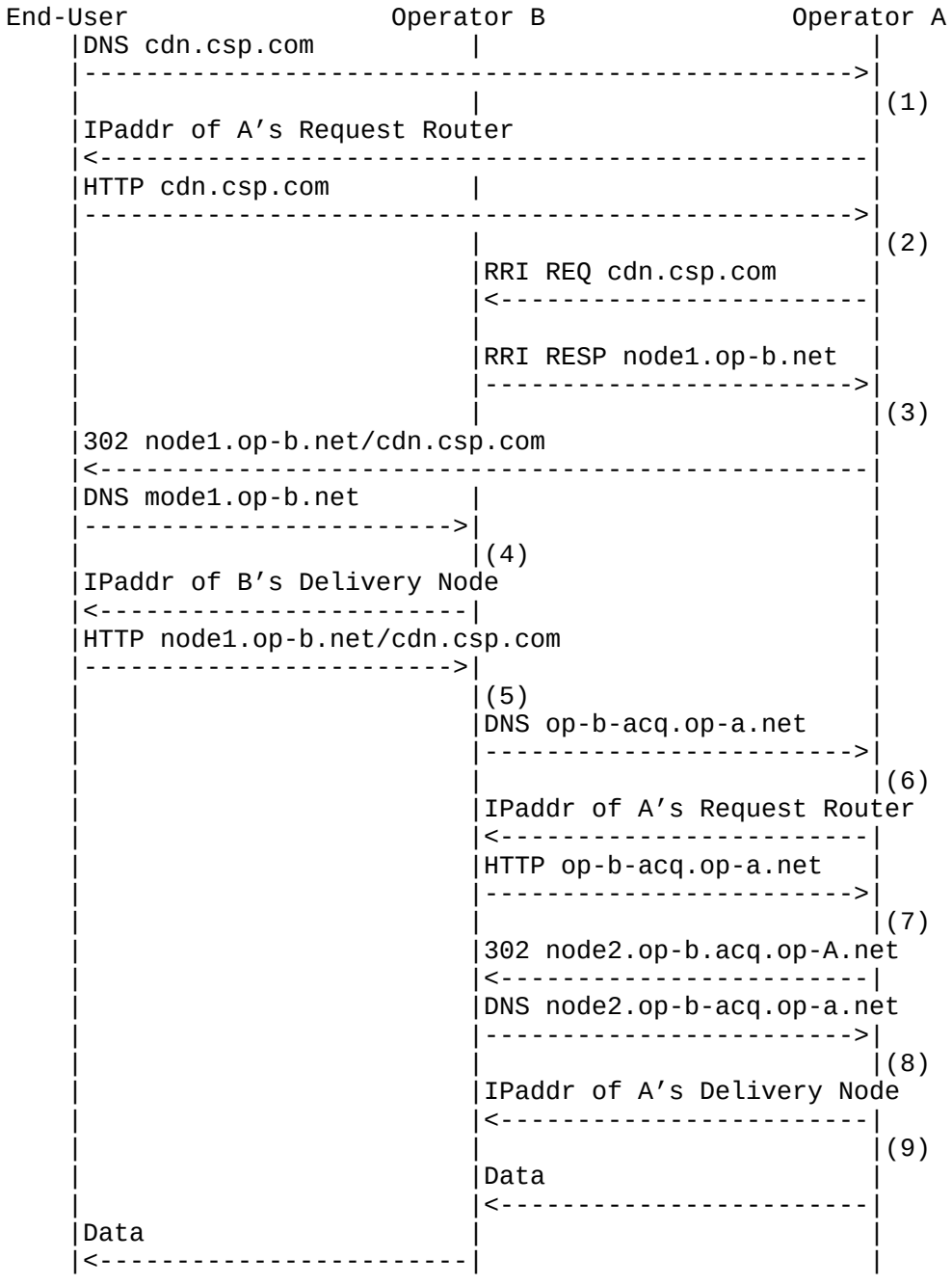


Figure 4: Request Trace for Recursive HTTP redirection method

The steps illustrated in the figure are as follows:

1. A DNS resolver for Operator A processes the DNS request for its customer based on CDN-domain cdn.csp.com. It returns the IP address of a Request Router in Operator A.

2. A Request Router for Operator A processes the HTTP request and recognizes that the end-user is best served by another CDN--specifically one provided by Operator B--and so it queries the CDNI Request Routing interface of Operator B, providing a set of information about the request including the URL requested. Operator B replies with the DNS name of a delivery node.

3. Operator A returns a 302 redirect message for a new URL obtained from the Request Routing Interface.

4. The end-user does a DNS lookup using the host name of the URL just provided (node1.op-b.net). B's DNS resolver returns the IP address of the corresponding delivery node. Note that, since the name of the delivery node was already obtained from B using the CDNI Request Routing Interface, there should not be any further redirection here (in contrast to the iterative method described above.)

5. The end-user requests the content from B's delivery node, potentially resulting in a cache miss. In the case of a cache miss, the content needs to be acquired from uCDN (not the CSP.) The distinguished CDN-domain op-b.net indicates to dCDN that this content is to be acquired from another CDN; stripping the CDN-domain reveals the original CDN-domain cdn.csp.com, dCDN may verify that this CDN-domain belongs to a known peer (so as to avoid being tricked into serving as an open proxy). It then does a DNS request for the inter-CDN Acquisition "distinguished" CDN-domain as agreed above (in this case, op-b-acq.op-a.net).

6. Operator A DNS resolver processes the DNS request and returns the IP address of a request router in operator A.

7. The request router for Operator A processes the HTTP request from Operator B delivery node. Operator A request router recognizes that the request is from a peer CDN rather than an end-user because of the dedicated inter-CDN acquisition domain (op-b-acq.op-a.net). (Note that without this specially defined inter-CDN acquisition domain, operator A would be at risk of redirecting the request back to operator B, resulting in an infinite loop). The request router for Operator A selects a suitable delivery node in uCDN to serve the inter-CDN acquisition request and returns a 302 redirect message for a new URL

constructed by replacing the hostname by a subdomain of the Operator A's distinguished inter-CDN acquisition domain that points to the selected delivery node.

8. Operator A recognizes that the DNS request is from a peer CDN rather than an end-user (due to the internal CDN-domain) and so returns the address of a delivery node. (Note that without this specially defined internal domain, Operator A would be at risk of redirecting the request back to Operator B, resulting in an infinite loop.)

9. Operator A serves content for the requested CDN-domain to dCDN. Although not shown, it is at this point that Operator A processes the rest of the URL: it extracts information identifying the origin server, validates that this server has been registered, and determines the content provider that owns the origin server. It may also perform its own content acquisition steps if needed before returning the content to dCDN.

3.3.1. Comments on the example

Recursive redirection has the advantage over iterative of being more transparent from the end-user's perspective, but the disadvantage of each CDN exposing more of its internal structure (in particular, the addresses of edge caches) to peer CDNs. By contrast, iterative redirection does not require dCDN to expose the addresses of its edge caches to uCDN.

This example happens to use HTTP-based redirection in both CDN A and CDN B, but a similar example could be constructed using DNS-based redirection in either CDN. Hence, the key point to take away here is simply that the end user only sees a single redirection of some type, as opposed to the pair of redirections in the prior (iterative) example.

The use of the Request Routing Interface requires that interface to be appropriately configured and bootstrapped, which is not shown here. More discussion on the bootstrapping of interfaces is provided in Section 4

3.4. DNS-based redirection example

In this section we walk through a simple example using DNS-based redirection for request redirection from uCDN to dCDN (as well as for request routing inside dCDN and uCDN). As noted in Section 2.1, DNS-based redirection has certain advantages over HTTP-based redirection (notably, it is transparent to the end-user) as well as some drawbacks (notably the client IP address is not visible to the

request router).

As before, Operator A must learn the set of requests that dCDN is willing or able to serve (e.g. which client IP address prefixes or geographic regions are part of the dCDN footprint). Operator B must have and make known to operator A some unique identifier that can be used for the construction of a distinguished CDN domain, as shown in more detail below. (This identifier strictly needs only to be unique within the scope of Operator A, but a globally unique identifier, such as an AS number assigned to B, is one easy way to achieve that.) Also, Operator A must obtain the NS records for Operator B's externally visible redirection servers. Also, as before, a distinguished CDN-domain, such as op-b-acq.op-a.net, must be assigned for inter-CDN acquisition.

DNS must be configured in the following way:

- The CSP must be configured to make Operator A the authoritative DNS server for cdn.csp.com (or to return a CNAME for cdn.csp.com for which operator A is the authoritative DNS server).

- When uCDN sees a request best served by dCDN, it returns CNAME and NS records for "b.cdn.csp.com", where "b" is the unique identifier assigned to Operator B. (It may, for example, be an AS number assigned to Operator B.)

- dCDN must be configured so that a request for "b.cdn.csp.com" returns a delivery node in dCDN.

- uCDN must be configured so that a request for "op-b-acq.op-a.net" returns a delivery node in uCDN.

Figure 5 depicts the exchange of DNS and HTTP requests. The main differences from Figure 3 are the lack of HTTP redirection and transparency to the end-user.

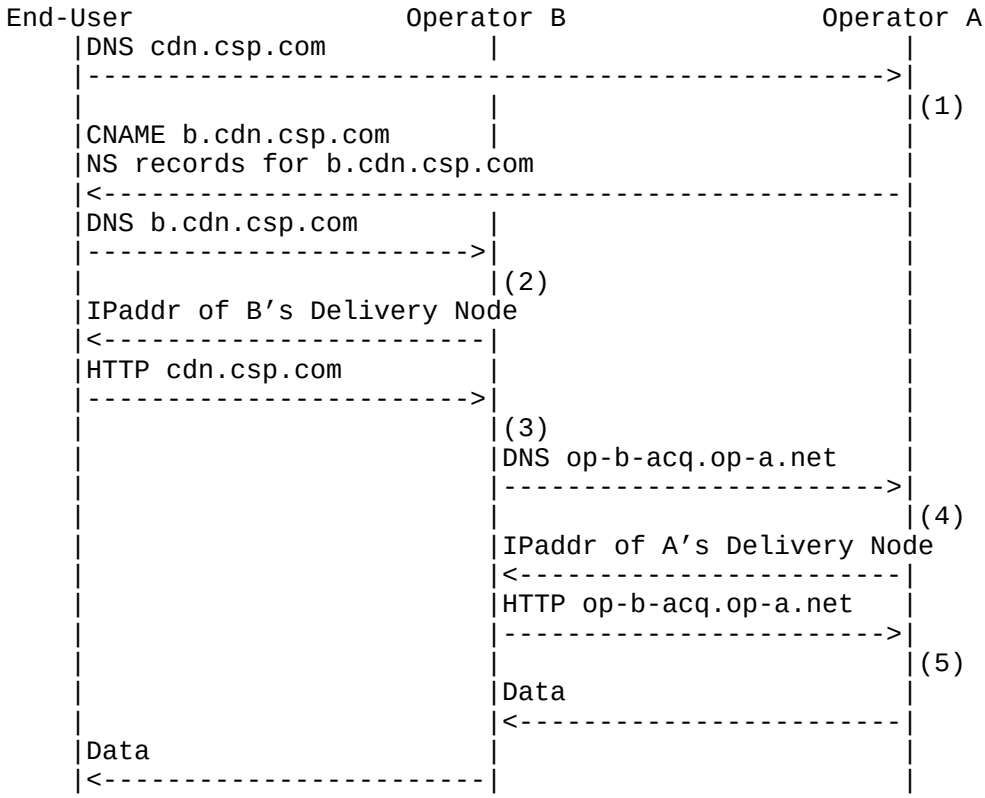


Figure 5: Request Trace for DNS-based Redirection Example

The steps illustrated in the figure are as follows:

1. Request Router for Operator A processes the DNS request for CDN-domain cdn.csp.com and recognizes that the end-user is best served by another CDN. (This may depend on the IP address of the user's local DNS resolver, or other information discussed below.) The Request Router returns a DNS CNAME response by "stacking" the distinguished identifier for Operator B onto the original CDN-domain (e.g., b.cdn.csp.com), plus an NS record that maps b.cdn.csp.com to B's Request Router.

2. The end-user does a DNS lookup using the modified CDN-domain (i.e., b.cdn.csp.com). This causes B's Request Router to respond with a suitable delivery node.

3. The end-user requests the content from B's delivery node. The requested URL contains the name cdn.csp.com. (Note that the returned CNAME does not affect the URL.) At this point the

delivery node has the correct IP address of the end-user and can do an HTTP 302 redirect if the redirections in steps 2 and 3 were incorrect. Otherwise B verifies that this CDN-domain belongs to a known peer (so as to avoid being tricked into serving as an open proxy). It then does a DNS request for an "internal" CDN-domain as agreed above (op-b-acq.op-a.net).

4. Operator A recognizes that the DNS request is from a peer CDN rather than an end-user (due to the internal CDN-domain) and so returns the address of a delivery node in uCDN.

5. Operator A serves content to dCDN. Although not shown, it is at this point that Operator A processes the rest of the URL: it extracts information identifying the origin server, validates that this server has been registered, and determines the content provider that owns the origin server.

3.4.1. Comments on the example

The advantages of this approach are that it is more transparent to the end-user and requires fewer round trips than HTTP-based redirection. A potential problem is that the upstream CDN depends on being able to learn the correct downstream CDN that serves the end-user from the client address in the DNS request. In standard DNS operation, uCDN will only obtain the address of the client's local DNS resolver (LDNS), which is not guaranteed to be in the same network (or geographic region) as the client. If not--e.g., the end-user uses a global DNS service--then the upstream CDN cannot determine the appropriate downstream CDN to serve the end-user. In this case, one option is for the upstream CDN to treat the end-user as it would any user not connected to a peer CDN. Another option is for the upstream CDN to "fall back" to a pure HTTP-based redirection strategy in this case (i.e., use the first method). Note that this problem affects existing CDNs that rely on DNS to determine where to redirect client requests, but the consequences are arguably less serious since the LDNS is likely in the same network as the dCDN serves. One approach to ensuring that the client's IP address prefix is correctly determined in such situations is described in [I-D.vandergaast-edns-client-subnet].

As with the prior example, this example partially illustrates the various interfaces involved in CDNI. Operator A could learn dynamically from Operator B the set of prefixes or regions that B is willing and able to serve via the request routing interface. The distinguished name used for acquisition and the identifier for Operator B that is prepended to the CDN domain on redirection are examples of information elements that might also be conveyed by CDNI interfaces (or, alternatively, statically configured). As before,

minimal metadata sufficient to obtain the content is carried "in-band" as part of the redirection process, and standard HTTP is used for inter-CDN acquisition. There is no explicit logging interface discussed in this example.

3.5. Dynamic Footprint Discovery

There could be situations where being able to dynamically discover the set of requests that a given dCDN is willing and able to serve is beneficial. For example, a CDN might at one time be able to serve a certain set of client IP prefixes, but that set might change over time due to changes in the topology and routing policies of the IP network. The following example illustrates this capability. We have chosen the example of DNS-based redirection, but HTTP-based redirection could equally well use this approach.

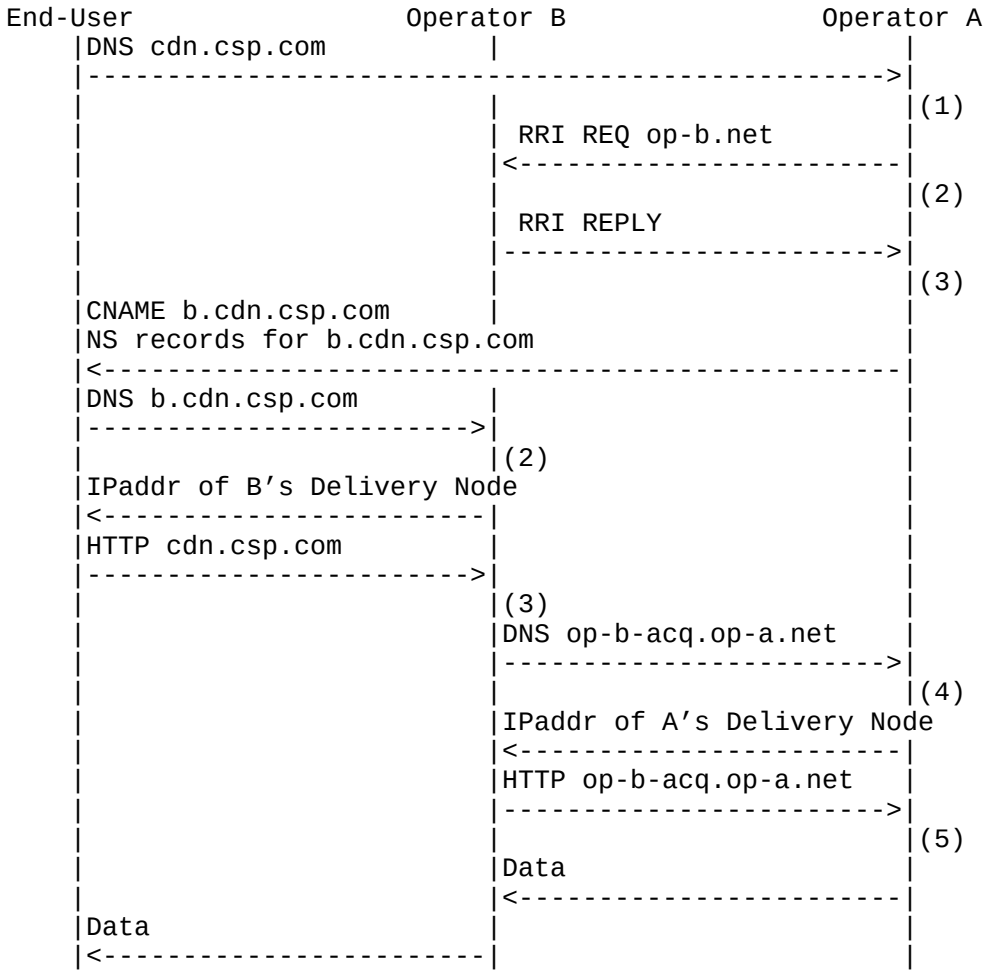


Figure 6: Request Trace for Dynamic Footprint Discovery Example

This example differs from the one in Figure 5 only in the addition of a CDNI Request Routing Interface request (step 2) and corresponding response (step 3). The RRI Req could be a message such as "Can you serve clients from this IP Prefix?" or it could be "Provide the list of client IP prefixes you can currently serve". In either case the response might be cached by operator A to avoid repeatedly asking the same question. Alternatively, or in addition, Operator B may spontaneously advertise to Operator A information (or changes) on the set of requests it is willing and able to serve on behalf of operator A; in that case, Operator B may spontaneously issue RRI REPLY messages that are not in direct response to a corresponding RRI REQ message. (Note that the issues of determining the client’s subnet

from DNS requests, as described above, are exactly the same here as in Section 3.4.)

Once Operator A obtains the RRI response, it is now able to determine that Operator B's CDN is an appropriate dCDN for this request and therefore a valid candidate dCDN to consider in its Redirection decision. If that dCDN is selected, the redirection and serving of the request proceeds as before (i.e. in the absence of dynamic footprint discovery).

3.6. Content Removal

The following example illustrates how the Metadata interface may be used to remove an item of content. In this example, user requests for a particular content, and corresponding redirection of such requests from Operator A to Operator B CDN, may (or may not) have taken place earlier. Then, at some point in time, the uCDN (for example, in response to a corresponding trigger from the Content Provider) uses the Metadata Interface to request that content identified by a particular URL be removed from dCDN. The following diagram illustrates the operation.

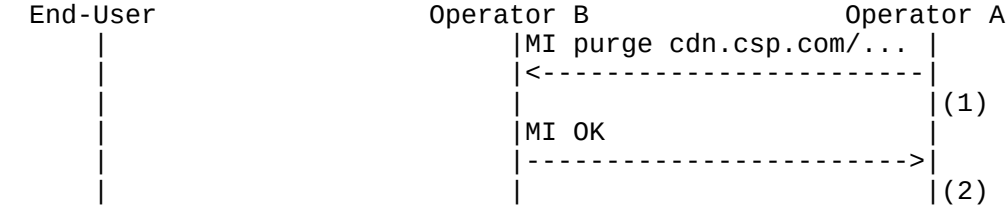


Figure 7: Request Trace for Content Removal

The metadata interface is used to convey the request from uCDN to dCDN that some previously acquired content should be deleted. The URL in the request specifies which content to remove. This example corresponds to a DNS-based redirection scenario such as Section 3.4. If HTTP-based redirection had been used, the URL for removal would be of the form peer-a.op-b.net/cdn.csp.com/...

The dCDN is expected to confirm to the uCDN, as illustrated by the MI OK message, the completion of the removal of the targeted content from all the caches in dCDN.

3.7. Pre-Positioned Content Acquisition Example

The following example illustrates how the metadata interface may be used to pre-position an item of content in the dCDN. In this example, Operator A uses the Metadata Interface to request that

content identified by a particular URL be pre-positioned into Operator B CDN.

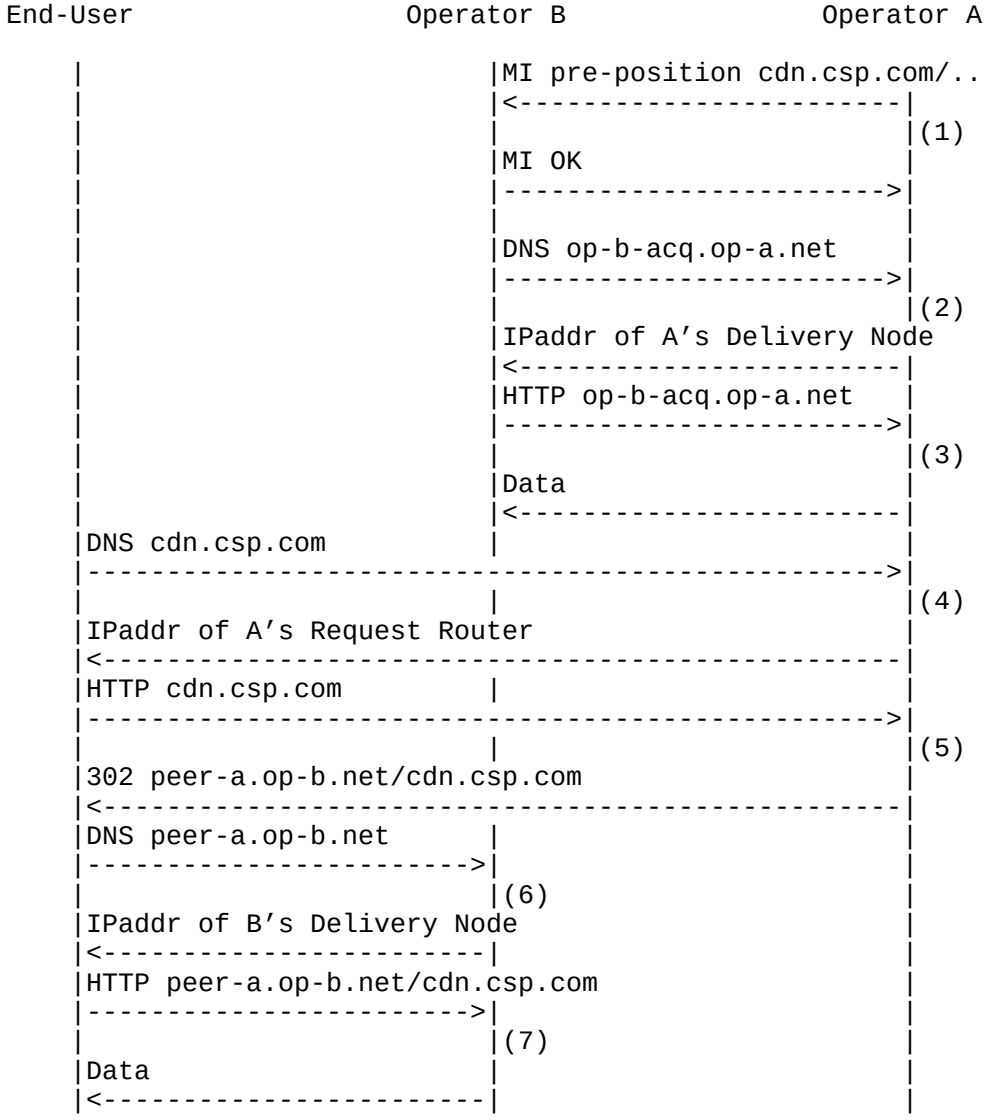


Figure 8: Request Trace for Content Pre-Positioning

The steps illustrated in the figure are as follows:

1. Operator A uses the Metadata Interface to request that Operator B pre-positions a particular content item identified by its URL.

Operator B responds by confirming that it is willing to perform this operation.

Steps 2 and 3 are exactly the same as steps 5 and 6 of Figure 3, only this time those steps happen as the result of the Pre-positioning request instead of as the result of a cache miss.

Steps 4, 5, 6, 7 are exactly the same as steps 1, 2, 3, 4 of Figure 3, only this time Operator B CDN can serve the end-user request without triggering dynamic content acquisition, since the content has been pre-positioned in dCDN. Note that, depending on dCDN operations and policies, the content pre-positioned in the dCDN may be pre-positioned to all, or a subset of, dCDN caches. In the latter case, intra-CDN dynamic content acquisition may take place inside the dCDN serving requests from caches on which the content has not been pre-positioning; however, such intra-CDN dynamic acquisition would not involve the uCDN.

3.8. Asynchronous CDNI Metadata Example

In this section we walk through a simple example illustrating a scenario of asynchronously exchanging CDNI metadata, where the downstream CDN obtains CDNI metadata for content ahead of a corresponding content request. The example that follows assumes that HTTP-based inter-CDN redirection and recursive CDNI request-routing are used, as in Section 3.3. However, asynchronous exchange of CDNI Metadata is similarly applicable to DNS-based inter-CDN redirection and iterative request routing (in which cases the CDNI metadata may be used at slightly different processing stages of the message flows).

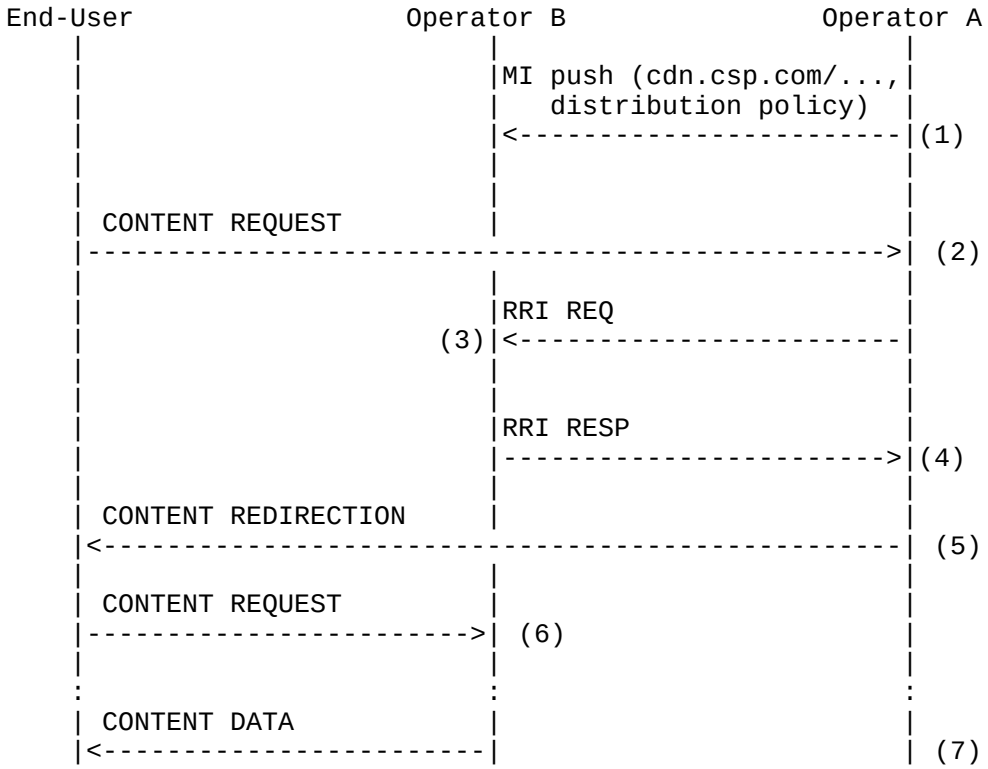


Figure 9: Request Trace for Asynchronous CDNI Metadata

The steps illustrated in the figure are as follows:

1. Operator A uses the Metadata Interface to asynchronously push CDNI metadata to Operator B. The present document does not constrain how the CDNI metadata information is actually represented. For the purposes of this example, we assume that Operator A provides CDNI metadata to Operator B indicating that:

   * this CDNI Metadata is applicable to any content referenced by "cdn.csp.com/op-b.net/..." (assuming HTTP redirection is used - it would be applicable to "cdn.csp.com/..." if DNS redirection were used as in Section 3.4).

   * this CDNI metadata consists of a distribution policy requiring enforcement by the delivery node of a specific per-request authorization mechanism (e.g. URI signature or token validation).

2. A Content Request occurs as usual.

3. A CDNI Request Routing Request (RRI REQ) is issued by operator A CDN, as discussed in Section 3.3. Operator B's request router can access the CDNI Metadata that are relevant to the requested content and that have been pre-positioned as per Step 1, which may or may not affect the response.

4. Operator B's request router issues a CDNI Request Routing Response (RRI RESP) as in Section 3.3.

5. Operator B performs content redirection as discussed in Section 3.3.

6. On receipt of the Content Request by the end user, the delivery node detects that previously acquired CDNI metadata is applicable to the requested content. In accordance with the specific CDNI metadata of this example, the delivery node will invoke the appropriate per-request authorization mechanism, before serving the content. (Details of this authorization are not shown.)

7. Assuming successful per-request authorization, serving of Content Data (possibly preceded by inter-CDN acquisition) proceeds as in Section 3.3.

3.9. Synchronous CDNI Metadata Acquisition Example

In this section we walk through a simple example illustrating a scenario of synchronous CDNI metadata acquisition, in which the downstream CDN obtains CDNI metadata for content at the time of handling a first request for the corresponding content. As in the preceding section, this example assumes that HTTP-based inter-CDN redirection and recursive CDNI request-routing are used (as in Section 3.3), but dynamic CDNI metadata acquisition is applicable to other variations of request routing.

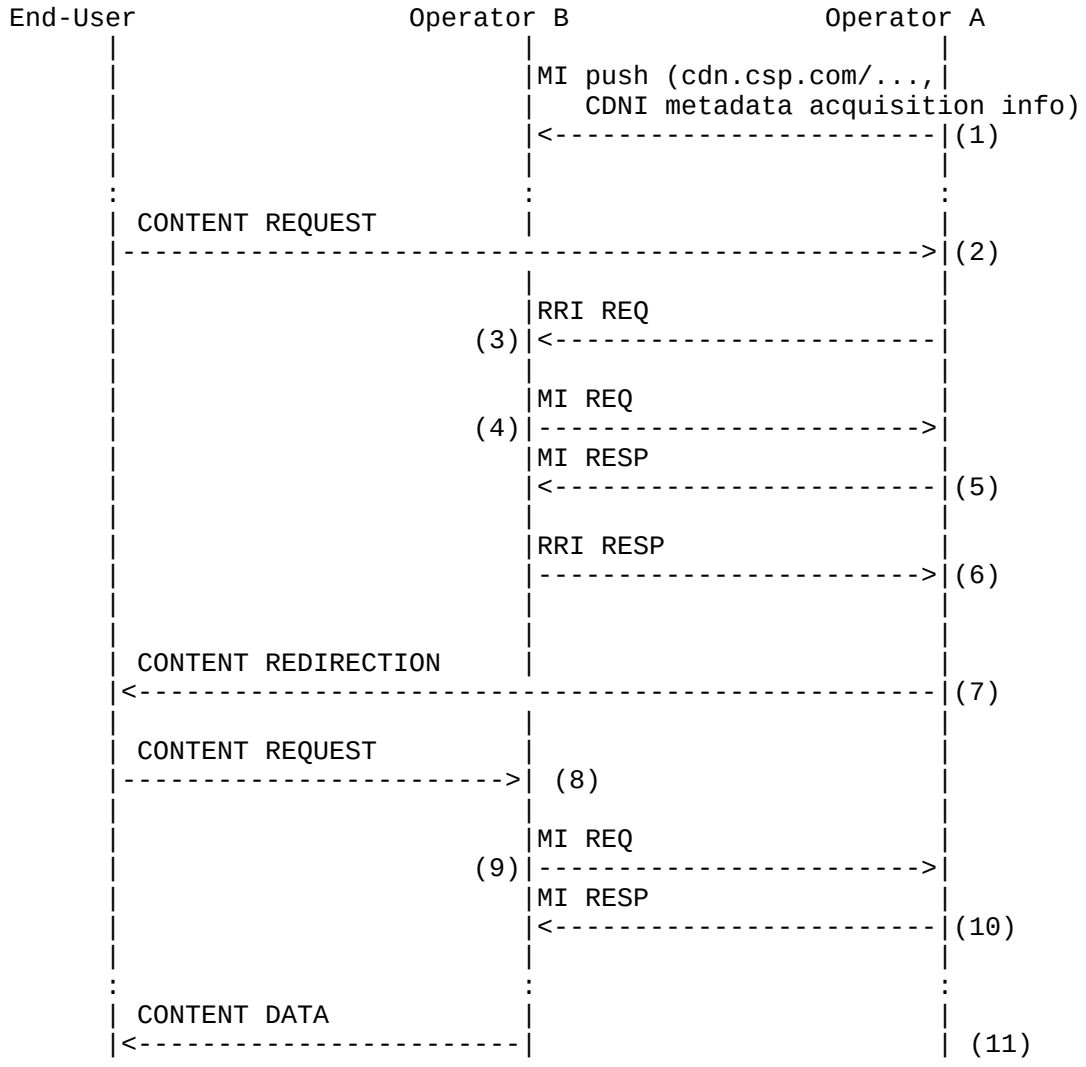


Figure 10: Request Trace for Synchronous CDNI Metadata Acquisition

The steps illustrated in the figure are as follows:

1. Operator A initially uses the Metadata Interface to asynchronously push seed metadata to Operator B. For example, this seed information may include a URI indicating where CDNI Metadata can later be pulled from for some content set. (There are alternative ways that this seeding information may be provided, such as piggybacking on the CDNI RRI REQ message of

Step 3.)

2. A Content Request arrives as normal.

3. A Request Routing Interface request occurs as in the prior example.

4. On receipt of the CDNI Request Routing Request, Operator B's CDN initiates synchronous acquisition of CDNI Metadata that are needed for routing of the end-user request. The seeding information provided in Step 1 is used to determine how to obtain the metadata. Note that there may exist cases in which this step does not occur (e.g., because the CDNI metadata seeding information indicates CDNI metadata are not needed at that stage).

5. On receipt of a CDNI Metadata MI Request, Operator A's CDN responds, making the corresponding CDNI metadata information available to Operator B's CDN. This metadata is considered by operator B's CDN before responding to the Request Routing request. (In a simple case, the metadata could simply be an allow or deny response for this particular request.)

6. Response to the RRI request as normal.

7. Redirection message is sent to the end user.

8. A delivery node of Operator B receives the end user request.

9. The delivery node triggers dynamic acquisition of additional CDNI metadata that are needed to process the end-user content request. Again the seeding information provided in Step 1 is used to determine how to acquire the needed CDNI metadata. Note that there may exist cases where this step need not happen, either because the metadata were already acquired previously, or because the seeding information indicates no metadata are required.

10. Operator A's CDN responds to the CDNI Metadata Request and makes the corresponding CDNI metadata available to Operator B. This metadata influence how Operator B's CDN processes the end-user request.

11. Content is served (possibly preceded by inter-CDN acquisition) as in Section 3.3.

4. Main Interfaces

Figure 1 illustrates the four main interfaces that are in scope for the CDNI WG, along with several others. The detailed specifications of these interfaces are left to other documents (mostly still to be written, but see [I-D.ietf-cdni-problem-statement] and [I-D.ietf-cdni-requirements] for some discussion of the interfaces).

One interface that is not shown in Figure 1 is the interface between the user and the CSP. While for the purposes of CDNI that interface is out of scope, it is worth noting that it does exist and can provide useful functions, such as end-to-end performance monitoring and some forms of authentication and authorization.

There is also an important interface between the user and the Request Routing function of both uCDN and dCDN. As we saw in some of the preceding examples, that interface can be used as a way of passing information such as the metadata that is required to obtain the content in dCDN from uCDN.

In this section we will provide an overview of the functions performed by each of the CDNI interfaces and discuss how they fit into the overall solution. We also examine some of the design tradeoffs. We begin with an examination of one such tradeoff that affects all the interfaces - the use of in-band or out-of-band communication.

4.1. In-Band versus Out-of-Band Interfaces

Before getting to the individual interfaces, we observe that there is a high-level design choice for each, involving the use of existing in-band communication channels versus defining new out-of-band interfaces.

It is possible that the information needed to carry out various interconnection functions can be communicated between peer CDNs using existing in-band protocols. The use of HTTP 302 redirect is an example of how certain aspects of request routing can be implemented in-band (embedded in URIs). Note that using existing in-band protocols does not imply that the CDNI interfaces are null; it is still necessary to establish the rules (conventions) by which such protocols are used to implement the various interface functions.

There are other opportunities for in-band communication beyond HTTP redirects. For example, many of the HTTP directives used by proxy servers can also be used by peer CDNs to inform each other of caching activity. Of these, one that is particularly relevant is the If-Modified-Since directive, which is used with the GET method to make

it conditional: if the requested object has not been modified since the time specified in this field, a copy of the object will not be returned, and instead, a 304 (not modified) response will be returned.

4.2. Request Routing Interface

We may think of the request routing interface as comprising two parts: the asynchronous advertisement of footprint and capabilities by a dCDN that allows a uCDN to decide whether to redirect particular user requests to that dCDN; and the synchronous operation of actually redirecting a user request. (These are somewhat analogous to the operations of routing and forwarding in IP.)

As illustrated in Section 3, the synchronous part of the request routing interface may be implemented in part by DNS and HTTP. Naming conventions may be established by which CDN peers communicate whether a request should be routed or content served.

In support of these exchanges, it is necessary for CDN peers to exchange additional information with each other. Depending on the method(s) supported, this includes

- The operator’s unique id (operator-id) or distinguished CDN-domain (operator-domain);
- NS records for the operator’s set of externally visible request routers;
- The set of requests the dCDN operator is prepared to serve (e.g. a set of client IP prefixes or geographic regions that may be served by dCDN).

Of these, the two operator identifiers are fixed, and can be exchanged off-line as part of a peering agreement. The NS records potentially change with some frequency, but an existing protocol--DNS--can be used to dynamically track this information. That is, a peer can do a DNS lookup on operator-domain to retrieve the set of NS records corresponding to the peer’s redirection service.

The set of requests that dCDN is willing to serve could in some cases be relatively static (e.g., a set of IP prefixes) which could be exchanged off-line, or might even be negotiated as part of a peering agreement. However, it may also be more dynamic, in which case an explicit protocol for its exchange would be be helpful.

A variety of options exist for the dCDN operator to advertise its footprint to uCDN. As discussed in

[I-D.previdi-cdni-footprint-advertisement], footprint is comprised of two components:

- a class of end user requests (represented, for example, by a set of IP prefixes, or a geographic region) that the dCDN is willing and able to serve directly, without use of another dCDN;

- the connectivity of the dCDN to other CDNs that may be able to serve content to users on behalf of dCDN.

[I-D.previdi-cdni-footprint-advertisement] describes an approach to advertising such footprint information asynchronously using BGP. In addition to this sort of information, a dCDN might also advertise "capabilities" such as the ability to handle certain types of content (e.g. specific streaming formats) or quality of service (QoS) capabilities. [I-D.xiaoyan-cdni-request-routing-protocol] describes an approach that exchanges CDN "capabilities" over HTTP, while [I-D.seedorf-alto-for-cdni] describes how ALTO [RFC5693] may be used to obtain request routing information.

We also note that the Request Routing interface plays a key role in enabling recursive redirection, as illustrated in Section 3.3. It enables the user to be redirected to the correct delivery node in dCDN with only a single redirection step (as seen by the user). This may be particularly valuable as the chain of interconnected CDNs increases beyond two CDNs.

4.3. Logging Interface

It is necessary for the upstream CDN to have visibility into the delivery of content it originates to end-users connected to the downstream CDN. This allows the upstream CDN to properly bill its customers for multiple deliveries of content cached by the downstream CDN, as well as to report accurate traffic statistics to those content providers. This is one role of the Logging interface.

Other operational data that may be relevant to CDNI can also be exchanged by the Logging interface. For example, dCDN may report the amount of content it has acquired from uCDN, and how much cache storage has been consumed by content cached on behalf of uCDN.

Traffic logs are easily exchanged off-line. For example, the following traffic log is a small deviation from the Apache log file format, where entries include the following fields:

- Domain - the full domain name of the origin server

- IP address - the IP address of the client making the request
- End time - the ending time of the transfer
- Time zone - any time zone modifier for the end time
- Method - the transfer command itself (e.g., GET, POST, HEAD)
- URL - the requested URL
- Version - the protocol version, such as HTTP/1.0
- Response - a numeric response code indicating transfer result
- Bytes Sent - the number of bytes in the body sent to the client
- Request ID - a unique identifier for this transfer
- User agent - the user agent, if supplied
- Duration - the duration of the transfer in milliseconds
- Cached Bytes - the number of body bytes served from the cache
- Referrer - the referrer string from the client, if supplied

Of these, only the Domain field is indirect in the downstream CDN--it is set to the CDN-domain used by the upstream CDN rather than the actual origin server. This field could then used to filter traffic log entries so only those entries matching the upstream CDN are reported to the corresponding operator.

One open question is who does the filtering. One option is that the downstream CDN filters its own logs, and passes the relevant records directly to each upstream peer. This requires that the downstream CDN knows the set of CDN-domains that belong to each upstream peer. If this information is already exchanged between peers as part of the request routing interface, then direct peer-to-peer reporting is straightforward. If it is not available, and operators do not wish to advertise the set of CDN-domains they serve to their peers, then the second option is for each CDN to send both its non-local traffic records and the set of CDN-domains it serves to an independent third-party (i.e., a CDN Exchange), which subsequently filters, merges, and distributes traffic records on behalf of each participating CDN operator.

A second open question is how timely traffic information should be. For example, in addition to off-line traffic logs, accurate real-time

traffic monitoring might also be useful, but such information requires that the downstream CDN inform the upstream CDN each time it serves upstream content from its cache. The downstream CDN can do this, for example, by sending a conditional HTTP GET request (If-Modified-Since) to the upstream CDN each time it receives an HTTP GET request from one of its end-users. This allows the upstream CDN to record that a request has been issued for the purpose of real-time traffic monitoring. The upstream CDN can also use this information to validate the traffic logs received later from the downstream CDN.

There is obviously a tradeoff between accuracy of such monitoring and the overhead of the downstream CDN having to go back to the upstream CDN for every request.

Another design tradeoff in the Logging interface is the degree of aggregation or summarization of data. One situation that lends itself to summarization is the delivery of HTTP-based adaptive bit-rate video. Most schemes to deliver such video use a large number of relatively small HTTP requests (e.g. one request per 2-second chunk of video.) It may be desirable to aggregate logging information so that a single log entry is provided for the entire video rather than for each chunk. Note however that such aggregation requires a degree of application awareness in dCDN to recognize that the many HTTP requests correspond to a single video.

Other forms of aggregation may also be useful. For example, there may be situations where bulk metrics such as bytes delivered per hour may suffice rather than the detailed per-request logs outlined above. It seems likely that a range of granularities of logging will be needed along with ways to specify the type and degree of aggregation required.

4.4. Control Interface

The control interface is primarily used for the bootstrapping of other interfaces. As a simple example, it could be used to provide the address of the logging server in dCDN to uCDN in order to bootstrap the logging interface. It may also be used, for example, to establish security associations for the other interfaces. We discuss the relationship between the Control and Metadata interfaces in the next section.

4.5. Metadata Interface

The role of the metadata interface is to enable CDNI distribution metadata to be conveyed to the downstream CDN by the upstream CDN. Such metadata includes geo-blocking restrictions, availability windows, access control policies, and so on. It may also include

policy information such as the desire to pre-position content rather than fetch it on demand.

Some metadata may be able to be conveyed using in-band mechanisms. For example, to inform the downstream CDN of any geo-blocking restrictions or availability windows, the upstream can elect to redirect a request to the downstream CDN only if that CDN's advertised delivery footprint is acceptable for the requested URL. Similarly, the request could be forwarded only if the current time is within the availability window.

Similarly, some forms of access control may also be performed on a per-request basis using HTTP directives. For example, being able to respond to a conditional GET request gives the upstream CDN an opportunity to influence how the downstream CDN delivers its content. Minimally, the upstream CDN can invalidate (purge) content previously cached by the downstream CDN.

Fine-grain control over how the downstream CDN delivers content on behalf of the upstream CDN is also possible. For example, by including the X-Forwarded-For HTTP header with the conditional GET request, the downstream CDN can report the end-user's IP address to the upstream CDN, giving it an opportunity to control whether the downstream CDN should serve the content to this particular end-user. The upstream CDN would communicate its directive through its response to the conditional GET. The downstream CDN can cache information for a period of time specified by the upstream CDN, thereby reducing control overhead.

Thinking beyond what metadata operations can be done in-line, we note that all CDNs already export a "content purge" operation to their customers. The CDNI metadata interface could support a similar "content purge" API call. When a CSP invokes purge on the upstream CDN, that CDN in turn invokes purge on all downstream CDNs that might be caching the content. Of course, agreement as to the syntax and semantics of this call is required.

One open question is how to distinguish between what functionality is supported by the Metadata interface and what functionality is supported by the Control interface. The approach taken in this document is to assume a minimal Control interface that is used to bootstrap the other interfaces. We assume all information that governs peer CDN behavior at the granularity of individual content items is exchanged via the Metadata interface. We note that some other documents have suggested that the purge operation should be part of the Control Interface. The authors' view is that purging a piece of content is just another form of metadata, similar to an availability window. In effect, a purge is equivalent to a statement

that the availability window for that content has now expired. The timeliness requirements for purge operations may affect the detailed design of the metadata interface.

5. Deployment Models

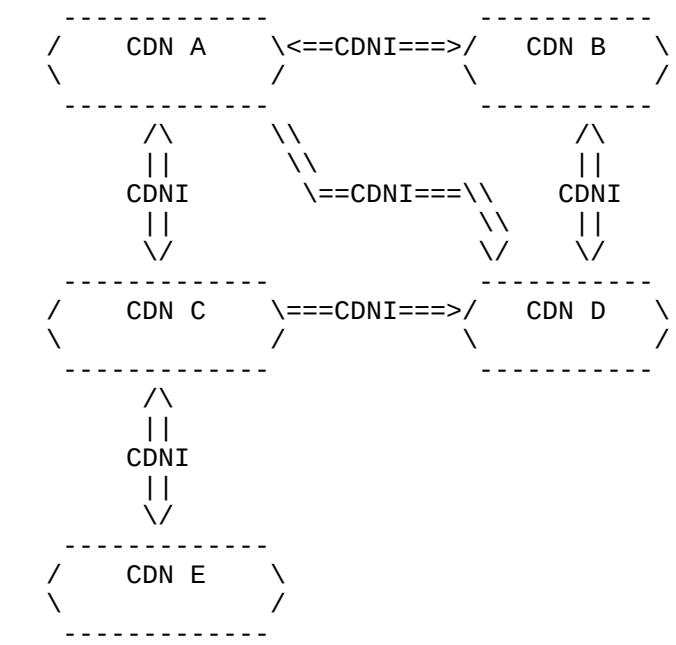
In this section we describe a number of possible deployment models that may be achieved using the CDNI interfaces described above. We note that these models are by no means exhaustive, and that may other models may be possible.

Although the reference model of Figure 1 shows all CDN functions on each side of the CDNI interface, deployments can rely on entities that are involved in any subset of these functions, and therefore only support the relevant subset of CDNI interfaces. As already noted in Section 3, effective CDNI deployments can be built without necessarily implementing all four interfaces. Some examples of such deployments are shown below.

Note that, while we refer to upstream and downstream CDNs, this distinction applies to specific content items and transactions. That is, a given CDN may be upstream for some transactions and downstream for others, depending on many factors such as location of the requesting client and the particular piece of content requested.

5.1. Meshed CDNs

Although the reference model illustrated in Figure 1 shows a unidirectional CDN interconnection with a single uCDN and a single dCDN, any arbitrary CDNI meshing can be built from this, such as the example meshing illustrated in Figure 11. (Support for arbitrary meshing may or may not be in the initial scope for the working group, but the model allows for it.)



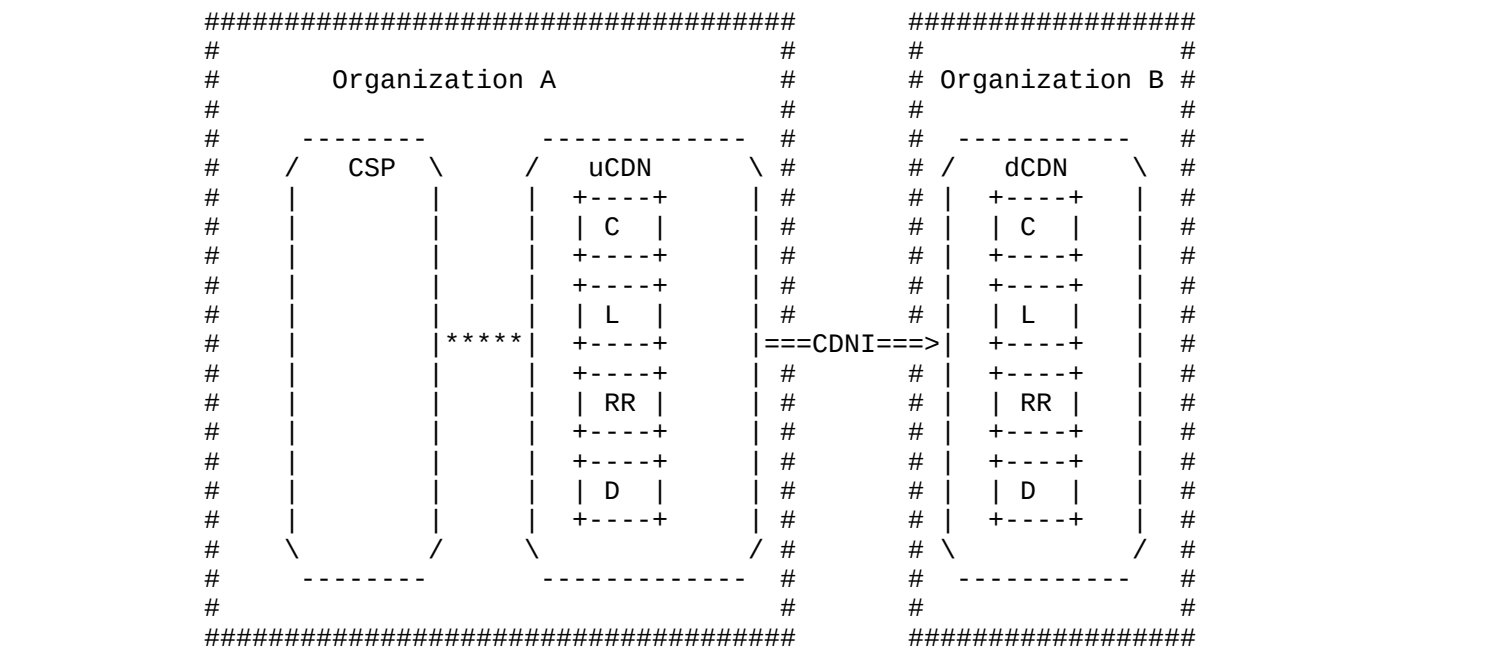
===> CDNI interfaces, with right-hand side CDN acting as dCDN to left-hand side CDN

<==> CDNI interfaces, with right-hand side CDN acting as dCDN to left-hand side CDN and with left-hand side CDN acting as dCDN to right-hand side CDN

Figure 11: CDNI Deployment Model: CDN Meshing Example

5.2. CSP combined with CDN

Note that our terminology refers to functional roles and not economic or business roles. That is, a given organization may be operating as both a CSP and a fully-fledged uCDN when we consider the functions performed, as illustrated in Figure 12.

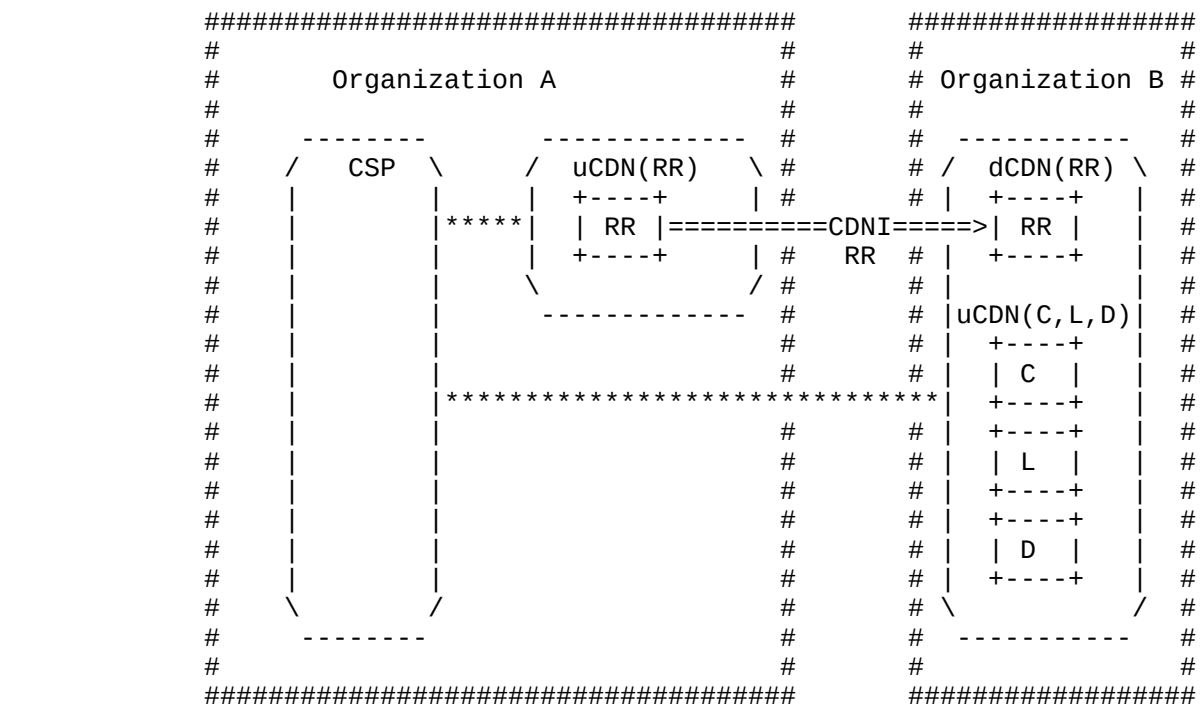


===> CDNI interfaces, with right-hand side CDN acting as dCDN to left-hand side CDN
**** interfaces outside the scope of CDNI
C Control component of the CDN
L Logging component of the CDN
RR Request Routing component of the CDN
D Distribution component of the CDN

Figure 12: CDNI Deployment Model: Organization combining CSP & uCDN

5.3. CSP using CDNI Request Routing Interface

As another example, a content provider organization may choose to run its own request routing function as a way to select among multiple candidate CDN providers; In this case the content provider may be modeled as the combination of a CSP and of a special, restricted case of a CDN. In that case, as illustrated in Figure 13, the CDNI Request Routing interface can be used between the restricted CDN operated by the content provider Organization and the CDN operated by the full-CDN organization acting as a dCDN in the request routing control plane. Interfaces outside the scope of the CDNI work can be used between the CSP functional entities of the content provider organization and the CDN operated by the full-CDN organization acting as a uCDN) in the CDNI control planes other than the request routing plane (i.e. Control, Distribution, Logging).



===> CDNI Request Routing interface
**** interfaces outside the scope of CDNI

Figure 13: CDNI Deployment Model: Organization combining CSP and partial CDN

5.4. CDN Federations and CDN Exchanges

There are two additional concepts related to, but distinct from CDN Interconnection. The first is CDN Federation. Our view is that CDNI is the more general concept, involving two or more CDNs serving content to each other's users, while federation implies a multi-lateral interconnection arrangement, but other CDN interconnection agreements are also possible (e.g., symmetric bilateral, asymmetric bilateral). An important conclusion is that CDNI technology should not presume (or bake in) a particular interconnection agreement, but should instead be general enough to permit alternative interconnection arrangements to evolve.

The second concept often used in the context of CDN Federation is CDN Exchange--a third party broker or exchange that is used to facilitate a CDN federation. Our view is that a CDN exchange offers valuable machinery to scale the number of CDN operators involved in a multi-

lateral (federated) agreement, but that this machinery is built on top of the core CDNI interconnection mechanisms. For example, as illustrated in Figure 14, the exchange might aggregate and redistribute information about each CDN footprint and capacity, as well as collect, filter, and re-distribute traffic logs that each participant needs for interconnection settlement, but inter-CDN request routing, inter-CDN content distribution (including inter-CDN acquisition) and inter-CDN control which fundamentally involve a direct interaction between an upstream CDN and a downstream CDN--operate exactly as in a pair-wise peering arrangement. Turning to Figure 14, we observe that in this example:

- each CDN supports a direct CDNI Control interface to every other CDN
- each CDN supports a direct CDNI Metadata interface to every other CDN
- each CDN supports a CDNI Logging interface with the CDN Exchange
- each CDN supports both a CDNI request Routing interface with the CDN Exchange (for aggregation and redistribution of dynamic CDN footprint discovery information) and a direct CDNI Request Routing interface to every other CDN (for actual request redirection).

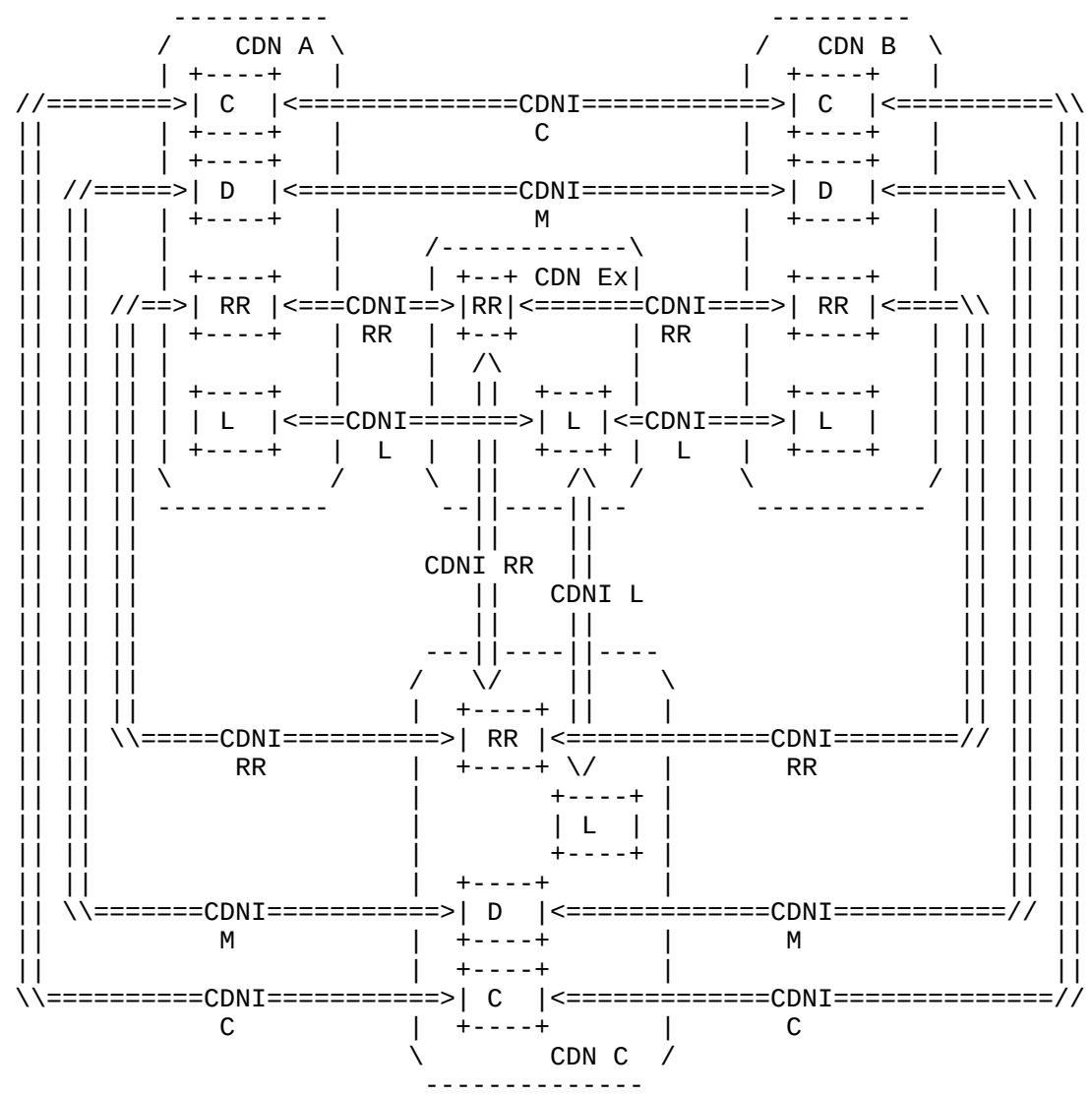


Figure 14: CDNI Deployment Model: CDN Exchange

Note that a CDN exchange may alternatively support a different set of functionality (e.g. Logging only, or Logging and full request

routing, or all the functionality of a CDN including content distribution). All these options are expected to be allowed by the IETF CDNI specifications.

6. Trust Model

There are a number of trust issues that need to be addressed by a CDNI solution. Many of them are in fact similar or identical to those in a simple CDN without interconnection. In a standard CDN environment (without CDNI), the CSP places a degree of trust in a single CDN operator to perform many functions. The CDN is trusted to deliver content with appropriate quality of experience for the end user. The CSP trusts the CDN operator not to corrupt or modify the content. The CSP often relies on the CDN operator to provide reliable accounting information regarding the volume of delivered content. The CSP may also trust the CDN operator to perform actions such as timely invalidation of content and restriction of access to content based on certain criteria such as location of the user and time of day, and to enforce per-request authorization performed by the CSP using techniques such as URI signing.

A CSP also places trust in the CDN not to distribute any information that is confidential to the CSP (e.g., how popular a given piece of content is) or confidential to the end user (e.g., which content has been watched by which user).

A CSP does not necessarily have to place complete trust in a CDN. A CSP will in some cases take steps to protect its content from improper distribution by a CDN, e.g. by encrypting it and distributing keys in some out of band way. A CSP also depends on monitoring (possibly by third parties) and reporting to verify that the CDN has performed adequately. A CSP may use techniques such as client-based metering to verify that accounting information provided by the CDN is reliable. HTTP conditional requests may be used to provide the CSP with some checks on CDN operation. In other words, while a CSP may trust a CDN to perform some functions in the short term, the CSP is able in most cases to verify whether these actions have been performed correctly and to take action (such as moving the content to a different CDN) if the CDN does not live up to expectations.

The main trust issue raised by CDNI is that it introduces transitive trust. A CDN that has a direct relationship with a CSP can now "outsource" the delivery of content to another (downstream) CDN. That CDN may in term outsource delivery to yet another downstream CDN, and so on.

The top level CDN in such a chain of delegation is responsible for ensuring that the requirements of the CSP are met. Failure to do so is presumably just as serious as in the traditional single CDN case. Hence, an upstream CDN is essentially trusting a downstream CDN to perform functions on its behalf in just the same way as a CSP trusts a single CDN. Monitoring and reporting can similarly be used to verify that the downstream CDN has performed appropriately. However, the introduction of multiple CDNs in the path between CSP and end user complicates the picture. For example, third party monitoring of CDN performance (or other aspects of operation, such as timely invalidation) might be able to identify the fact that a problem occurred somewhere in the chain but not point to the particular CDN at fault.

In summary, we assume that an upstream CDN will invest a certain amount of trust in a downstream CDN, but that it will verify that the downstream CDN is performing correctly, and take corrective action (including potentially breaking off its relationship with that CDN) if behavior is not correct. We do not expect that the trust relationship between a CSP and its "top level" CDN will differ significantly from that found today in single CDN situations. However, it does appear that more sophisticated tools and techniques for monitoring CDN performance and behavior will be required to enable the identification of the CDN at fault in a particular delivery chain.

We expect that the detailed designs for the specific interfaces for CDNI will need to take the transitive trust issues into account. For example, explicit confirmation that some action (such as content removal) has taken place in a downstream CDN may help to mitigate some issues of transitive trust.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

While there is a variety of security issues introduced by a single CDN, we are concerned here specifically with the additional issues that arise when CDNs are interconnected. For example, when a single CDN has the ability to distribute content on behalf of a CSP, there may be concerns that such content could be distributed to parties who are not authorized to receive it, and there are mechanisms to deal with such concerns. Our focus in this section is on how CDN interconnection introduces new security issues not found in the

single CDN case.

Many of the security issues that arise in CDNI are related to the transitivity of trust (or lack thereof) described in Section 6. As noted above, the design of the various interfaces for CDNI must take account of the additional risks posed by the fact that a CDN with whom a CSP has no direct relationship is now potentially distributing content for that CSP. The mechanisms used to mitigate these risks may be similar to those used in the single CDN case, but their suitability in this more complex environment must be validated.

Another concern that arises in any CDN is that information about the behavior of users (what content they access, how much content they consume, etc.) may be gathered by the CDN. This risk certainly exists in inter-connected CDNs, but it should be possible to apply the same techniques to mitigate it as in the single CDN case.

CDNs today offer a variety of means to control access to content, such as time-of-day restrictions, geo-blocking, and URI signing. These mechanisms must continue to function in CDNI environments, and this consideration is likely to affect the design of certain CDNI interfaces (e.g. metadata, request routing.)

Just as with a single CDN, each peer CDN must ensure that it is not used as an "open proxy" to deliver content on behalf of a malicious CSP. Whereas a single CDN typically addresses this problem by having CSPs explicitly register content (or origin servers) that is to be served, simply propagating this information to peer downstream CDNs may be problematic because it reveals more information than the upstream CDN is willing to specify. (To this end, the content acquisition step in the earlier examples force the dCDN to retrieve content from the uCDN rather than go directly to the origin server.)

There are several approaches to this problem. One is for the uCDN to encoded a signed token generated from a shared secret in each URL routed to a dCDN, and for the dCDN to validate the request based on this token. Another one is to have each upstream CDN advertise the set of CDN-domains they serve, where the downstream CDN checks each request against this set before caching and delivering the associated object. Although straightforward, this approach requires operators to reveal additional information, which may or may not be an issue.

8.1. Security of CDNI Interfaces

It is noted in [I-D.ietf-cdni-requirements] that all CDNI interfaces must be able to operate securely over insecure IP networks. Since it is expected that the CDNI interfaces will be implemented using existing application protocols such as HTTP or XMPP, we also expect

that the security mechanisms available to those protocols may be used by the CDNI interfaces. Details of how these interfaces are secured will be specified in the relevant interface documents.

8.2. Digital Rights Management

Issues of digital rights management (DRM, also sometimes called digital restrictions management) is often employed for content distributed via CDNs. In general, DRM relies on the CDN to distribute encrypted content, with decryption keys distributed to users by some other means (e.g. directly from the CSP to the end user.) For this reason, DRM is considered out of scope for the CDNI WG [I-D.ietf-cdni-problem-statement] and does not introduce additional security issues for CDNI.

9. Contributors

The following individuals contributed to this document:

- Francois le Faucheur
- Ben Niven-Jenkins
- David Ferguson
- John Hartman

10. Acknowledgements

We thank Aaron Falk and Huw Jones for their helpful input to the draft.

11. Informative References

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-01 (work in progress), October 2011.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-01 (work in progress), October 2011.

[I-D.ietf-cdni-use-cases]
Bertrand, G., Emile, S., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

[I-D.previdi-cdni-footprint-advertisement]
Previdi, S., Faucheur, F., Faucheur, L., and J. Medved, "CDNI Footprint Advertisement", draft-previdi-cdni-footprint-advertisement-00 (work in progress), October 2011.

[I-D.seedorf-alto-for-cdni]
Seedorf, J., "ALTO for CDNi Request Routing", draft-seedorf-alto-for-cdni-00 (work in progress), October 2011.

[I-D.vandergaast-edns-client-subnet]
Contavalli, C., Gaast, W., Leach, S., and D. Rodden, "Client subnet in DNS requests", draft-vandergaast-edns-client-subnet-00 (work in progress), January 2011.

[I-D.xiaoyan-cdni-request-routing-protocol]
He, X., Li, J., Dawkins, S., and G. Chen, "Request Routing Protocol for CDN Interconnection", draft-xiaoyan-cdni-request-routing-protocol-00 (work in progress), October 2011.

[RFC3466] Day, M., Cain, B., Tomlinson, G., and P. Rzewski, "A Model for Content Internetworking (CDI)", RFC 3466, February 2003.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

Authors' Addresses

Bruce Davie (editor)
Cisco Systems, Inc.
1414 Mass. Ave.
Boxborough, MA 01719
USA

Email: bsd@cisco.com

Larry Peterson (editor)
Verivue, Inc.
2 Research Way
Princeton, NJ
USA

Phone: +1 978 303 8032
Email: lpeterson@verivue.com

Internet Engineering Task Force
Internet Draft
Intended Status: Informational
Expires: April 26, 2012

Th. Zahariadis, Ed.
Synelixis
Y. Le Louedec
Orange Labs
Ch. Timmerer
UNI-KLU
S. Spirou
Intracom
D. Griffin
UCL

October 24, 2011

Catalogue of Advanced Use Cases for Content Delivery Network Interconnection

draft-fmn-cdni-advanced-use-cases-00

Abstract

The purpose of this draft is to complement the current CDNi WG use-cases with a catalogue of advanced, longer-term CDN interconnection use cases. The work has been the contribution of six European Commission (EC) co-funded projects, which are part of the EC Future Media networks (FMN) cluster.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Table of Contents

1 Introduction

The purpose of this draft is to complement the current CDNi Work Group short-term use-cases with a catalogue of advanced, longer-term CDN interconnection use cases. The proposed catalogue of use cases is coming from or inspired by work in the European Commission (EC) Future Media Network (FMN) cluster research projects. Though they are beyond the current short-term objectives of the CDNi WG, the proposed use cases could drive future work in the CDNi WG, especially in case of WG re-chartering (once the present goals will be reached). The use cases are derived from ongoing research work in six EC co-funded projects where concrete design, implementation and evaluation work is being undertaken to validate the approaches.

Moreover, this draft compares CDNs with Information-Centric Networking (ICN) which have similar goals and use cases. We discuss here this relation for the benefit of people and organizations working in these areas.

1.1 Terminology

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 [RFC2119].

The present document reuses terminology defined by CDNi WG documents [I-D.ietf-cdni-problem-statement], [I-D.ietf-cdni-use-cases] and [I-D.ietf-cdni-requirements]".

1.2 Abbreviations

[Ed. Note: List of abbreviations to be updated later]

- CSP: Content Service Provider
- dCDN: downstream CDN
- FMN: Future Media Networks
- ICN: Information-Centric Networking
- NSP: Network Service Provider
- QoE: Quality of Experience
- SLA: Service Level Agreement
- uCDN: upstream CDN

- MPD: Media Presentation Description
- DASH: Dynamic Adaptive Streaming over HTTP

2 Advanced/Long-term CDNi Use Cases

This draft includes four advanced CDNi use cases. The first use case introduces a Caching-CDN interconnection that put emphasis on the in-the network caching mechanism. As CDNi WG mainly targets interconnection between two CDNs, the second use case extends this use case to large-scale CDNs. The third use case introduces support of dynamic adaptive streaming over HTTP (DASH) in a multi-CDNs context, which may be today's most promising video distribution method as it overcomes limitations posed by firewalls and promises efficient distribution utilizing CDNs and network caching resources. Finally, the forth use case proposes the dynamic expansion of CDN capacity and geographical reach.

2.1 Use Case 1: Caching-CDN interconnection

Some telecom operators and NSP deploy caching servers, a.k.a. "transparent caching" servers, in their IP networks in order to cache content by CDNs over Internet, with as goal to relax and balance the load on their IP networks.

Today in most situations the two overlay networks - the upstream CDN (uCDN) and the downstream CDN (dCDN) set of transparent caching servers - run independently from each other. There is no specific interconnection interface between the two overlay networks.

There could be some interest to set up interfaces between these overlay networks (of course on condition that their owners get the right business incentives for).

Here are two illustrations of the potential benefits (this is not an exhaustive list):

- Setting up a "logging interface" between the two overlay networks could be beneficial for providing the uCDN (and beyond the Content Service Providers, CSP) with useful logging, monitoring, reporting data.

- Setting up a "CDN metadata interface" between the two overlay networks could be beneficial for allowing the uCDN to request that a given content file be purged from, or invalidated in, any downstream caching server.

The current charter of the CDNi WG states "the WG will not define

support for transparent caching across CDNs". The first priority is indeed to meet the goals and milestones specified in the current charter.

This use case proposal aims at recommending that this "caching-cdn interconnection" use case be considered in case of WG re-chartering (once the present goals will be reached).

The first difference with the present cdn-cdn interconnection model addressed by the CDNi WG is that there is no request routing interface in this "caching-cdn interconnection" use case. Then different sub-cases can be envisioned, depending on which CDNi interfaces are leveraged (with or without logging interface, with or without CDNI metadata interface, etc.). In a first approach, this "caching-cdn interconnection" use case could therefore be considered as a sub-case of the current CDNi WG's cdn-cdn interconnection model.

Note. Once such specific interfaces are set up between the uCDN and the dCDN set of transparent caching servers, the latter ones cannot be any more considered as fully transparent, at least from the viewpoint of the uCDN. This should call for a slight evolution of the terminology.

2.2 Use Case 2: CDN-CDN interconnections at large scale

The current focus of the CDNi WG is on the interconnection between two CDNs.

The purpose here would be to investigate situations where (possibly much) more than two CDNs are involved in a CDN federation.

As stated by the Amplification Principle defined in [RFC3439] "there do exist non-linearities which do not occur at small to medium scale, but occur at large scale". As a result, the number of involved CDNs could impact on the requirements and constraints, especially in terms of scalability, and therefore on the conceivable technical options to ensure interconnection.

As an illustration of the amplification principle application in CDNi, the initial considerations, and potential issues, about request routing in CDN interconnect scenarios presented in [I-D.stiemerling-cdni-routing-cons] could be even more critical in large scale CDN federations.

This would possibly lead to the definition of specific sub cases corresponding to cdn-cdn interconnections at large scale. Yet the

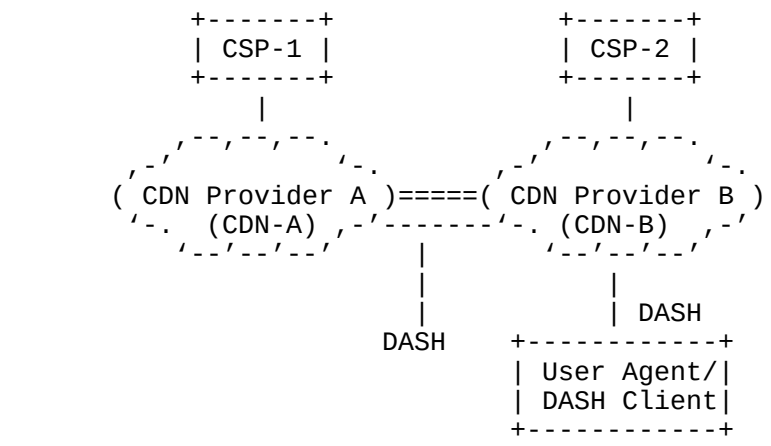
first priority (and prerequisite before investigating such large scale use cases) is to meet the goals and milestones specified in the current charter, i.e. to specify adequate interfaces for interconnecting two CDNs. So this proposal aims at recommending that this "cdn-cdn interconnections at large scale" use case be considered in case of WG re-chartering (once the present goals will be reached).

2.3 Use Case 3: Dynamic adaptive streaming over HTTP in multi-CDNs

The ever-growing video traffic on the Internet requires more efficient content distribution mechanisms. Compared to existing RTP/RTSP-based streaming or HTTP progressive download, Dynamic Adaptive Streaming over HTTP (DASH) enables stateless communication between a client and the corresponding server, better content adaptability and support for live media services [Stockhammer2011]. Intrinsic characteristic of DASH is the content fragmentation into various representations comprising segments (or sub-segments) of different encodings of one or several media components. These components/segments are transferred, along with content metadata descriptors referred to as media presentation description (MPD), to the origin servers. Using MPD, clients request the segments utilizing HTTP GET or partial GET requests.

DASH is considered as a promising solution for efficient and high-quality delivery of streaming services over the Internet and is currently standardized as 3GP-DASH, MPEG-DASH, and OIPF HAS. It supports among others, re-use of existing technologies (codecs, containers, etc.), deployment on top of HTTP-CDNs, re-use of HTTP origin and cache/proxy servers, re-use of existing media play-out engines, as well as on-demand, live and time-shifted delivery.

For example, DASH may be exploited within specific CDNs enabling clients to retrieve content hosted by given surrogates, taking into account the scale, the coverage and the reliability of HTTP-based CDN systems. Additionally, DASH can be also utilized as a delivery solution in a CDN Interconnect environment, enabling content acquisition among different providers. In such a case, a content service provider (e.g., CSP-1 in the following figure) will first ingest the prepared content into CDN-A, so that each surrogate can act as a DASH server offering the streaming service to the requesting End-User, acting as DASH client, connected with a different CDN (e.g., CDN-B). Towards this, CDN Interconnection requires interfaces among CDNs to be capable of facilitating their collaboration besides ensuring content streaming efficiently.



=== CDN Interconnection

In other words, in an interconnected CDN environment the definition of a control interface is required, which will enable DASH clients to acquire specific information from a hosting CDN over multiple-CDNs. Towards these, the anticipated interface must be able of providing/delivering:

- The Media Presentation Description that contains metadata to construct appropriate HTTP-URLs to access segments and to provide the streaming service to the user.

- The Number of source surrogates: one or multiple (content segments can be acquired from multiple sources).

- The location of source surrogates: e.g., locality-aware capabilities for choosing the nearest source(s), as a matter of geographical (internal or external) or network-based metrics (e.g., using latency or cost-based metrics).

- Delivery protocols: Definition of the delivery protocols used for the delivery of segments.

- Additional metadata for content delivery (e.g., metadata for content-aware networks).

2.4 Use Case 4: Dynamic expansion of CDN capacity and geographical reach

ISPs may offer a set of specialized network services which may be invoked by their customers, including CDN providers, with appropriate prior agreements and possible payments. The network service of most relevance to this use case is the provision of caching resources located within the ISP. A CDN provider making use of such a service

may invoke new caching resources within a local or remote ISP to dynamically create new CDN surrogate nodes. The newly created resources are provided by the network provider but under the control of the CDN provider.

This is an alternative way of dealing with increased load on a CDN, and a CDN provider who is able to invoke dynamic nodes is therefore able to expand dynamically to accommodate increased traffic demand or extend geographical reach. Such a CDN provider could a) advertise its elasticity to upstream CDNs which could simplify/enhance its resource-routing algorithm decisions, or b) advertise its new capabilities dynamically as it expands/contracts. These announcements could be made part of the CDNi control interface interactions and contributions could be made on the protocol extensions to announce capabilities dynamically and also to propose elasticity metrics.

3 Relationship between CDNI and Information-Centric Networking

CDN interconnection has similar goals and use cases to Information-Centric Networking (ICN). We discuss here this relation for the benefit of people and organizations working in these areas. We start with a very brief description of CDNs and ICN in order to have a common understanding. We then discuss similarities and differences in terms of objectives, technical approach, deployment and business models. Finally, we explore the possibility of interaction and coexistence of ICN and CDN in a future Internet. CDNs are real working systems, while ICN is still at the research stage. It is important to note that our analysis is based on the state of ICN research and the assumption that ICN will meet its design goals.

A CDN is a privately owned overlay network that aims to optimize delivery of content from (Content Service Providers) CSPs to End Users. CDNs are independent of each other. Optimization is in terms of performance, availability and cost. CDNs operate by serving content to User Agents from managed caches - called surrogates - at the edge of the network. CDNs may also use reserved network resources. The CDN provider collaborates with NSPs on surrogate placement and network resource reservation. The deployment, extension and (part of) the operation of CDNs are centrally managed. To maintain transparency for End Users, normal content locators (e.g., URLs) are used to access content. However, the CDN treats those locators as simple content identifiers when selecting a surrogate, in effect, decoupling the locator from the content location.

ICN shares many of the goals of CDNs. Optimization of delivery with ICN usually entails caching, QoS-aware routing and traffic differentiation, to various degrees. Unlike CDNs, ICN aims at

operating natively at the NSP level (although operation as an overlay is also possible). An NSP deploys ICN-enabled elements (mainly routers) with minimal planning in terms of content demand. The ICN reach grows with each new ICN-enabled NSP, without any central management. For content access, ICN uses explicit content identifiers that are independent from the content location.

A main objective of both CDNs and ICN is the effective delivery of content from CSPs to End Users. ICN also aims at accommodating End Users as small CSPs, i.e., End Users who want to share content. The decoupling of content identifiers from the content location is an explicit goal in ICN, while in CDN this is a by-product.

The main elements of a CDN are a set of content servers and a request redirection system. The content servers are carefully placed to be close to the User Agents and can be populated prior to any requests based on foreseen demand. An End User requests content with a normal URL, which triggers a part of the redirection system that resides at the seeming location of the content. The redirection system selects a content server based on criteria aiming to optimize some aspect of the delivery task. The selected content server then delivers the requested content to the User Agent.

In ICN, any edge router with storage capabilities can act as a content server, which is populated based on actual demand. Alternatively, or in addition, QoS-aware routing and traffic differentiation techniques are used to establish a path from the content source to the User Agent. Content is requested with a dedicated identifier. This identifier can be used to route the request to the content source or to an intermediate content server, while constructing the path. Alternatively, the identifier can be used as an index in a directory system to retrieve content metadata. The metadata are then used to setup a path from the content source to the User Agent.

A CDN constitutes an administrative domain, whereas in ICN an administrative domain can be as granular as an ICN router. When viewed in terms of administrative domains, CDN interconnection resembles the interconnection of ICN elements/domains. As such, request routing in CDNI and in ICN presents similar technical challenges. Following this thinking, the Content Distribution Metadata and CDNI Metadata of CDNI become related to the content metadata that are used in ICN to establish a path.

A CDN is deployed as an overlay with all its elements independent from NSPs and belonging to the same CDN Provider. The CDN Provider works closely with the CSP in order to ingest and place content. There’s also close collaboration between the CDN Provider and the

NSPs for server placement and reservation of resources. Once the CDN is operational, any unforeseen change in CSP requirements, network conditions or End User demand will force the CDN Provider to re-design the deployment. To avoid this, CDNs are designed with over-provisioning and redundancy. On the other hand, ICN is deployed as NSP infrastructure. Each NSP owns and independently manages the ICN elements in its domain. There's no "ICN Provider" to oversee the deployment of ICN. The system grows as each NSP becomes ICN-enabled. In order to use ICN, CSPs need to provide content source servers and content metadata. ICN is designed with flexibility in mind, which permits coping with many unforeseen events.

In a typical business case for CDNs, a CDN is deployed in and NSP domain and the NSP acts both as a CDN Provider and a (sole) CSP. NSPs use this model to offer content services, such as IPTV, to their End Users and so to differentiate their business. In another model for CDNs, a CDN Provider agrees with several NSPs to deploy content servers and reserve resources in their domain. The CDN Provider then sells content delivery services to interested CSPs.

The ICN business model is very similar to that for connectivity services. An NSP deploys ICN in its domain and makes transit or peering agreements with other ICN-enabled NSPs. The NSP then offers content delivery services to CSPs. It is also possible for the NSP to offer content consumption services to its End Users.

In a future Internet where ICN is deployed by some NSPs, it would be difficult to meet end-to-end the ICN performance goals, because of the "holes" between ICN domains. In such a case, CDNs can act as overlay bridges, because they might already have content close to the downstream ICN domain. In essence, from the ICN point of view, CDN Providers would become CSPs. However, the incentive for the CDN Provider is unclear, as such a move could undermine its own content delivery service. Borrowing from the motivation for CDNI, a CDN Provider who wants to extend its reach, instead of interfacing with a neighboring CDN, could interface with an ICN-enabled NSP. This is of course a scenario that only market forces and time can validate.

4 Acknowledgements

This draft has been produced by the Future Media Networks (FMN) cluster of the Networked Media Systems FP7 projects. The work leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) in(alphabetic order):
ALICANTE (FP7-ICT-248652; http://www.ict-alicante.eu/),
COAST (FP7-ICT-248036; http://www.coast-fp7.eu/),
COMET (FP7-ICT-248784; http://www.comet-project.org/),
ENVISION (FP7-ICT-248565; http://www.envision-project.org/),

NEXTMEDIA (FP7-ICT-249065; http://www.fi-nextmedia.eu/) and
OCEAN (FP7-ICT-248775; http://www.ict-ocean.eu/).

4.1 List of Contributors

Andrzej Beben, Warsaw University of Technology, Poland;

Christian Timmerer, UNI-KLU, Austria;

David Florez Rodriguez, Telefonica R&D, Spain;

David Griffin, Yiannis Psaras, Raul Landa, UCL, UK;

Daniel Negru, Labri, France;

Evangelos Pallis, Petros Anapliotis, TEIC, Greece;

George Xilouris, DEMOKRITOS, Greece;

Isidro Laso, European Commission, Belgium (on a personal basis);

Klaus Satzke, Alcatel-Lucent Bell Labs, Germany;

Michalis Georgiadis, PrimeTel, Cyprus;

Ning Wang, University of Surrey, UK;

Spiros Spirou, Intracom Telecom, Greece;

Theodore Zahariadis (editor), Synelixis, Greece;

Yannick Le Louedec, Orange Labs, France;

Yiping Chen, Daniel Negru, CNRS-LaBRI, France.

5 References

5.1 Normative References

[RFC3439] R. Bush, D. Meyer, "Internet Architectural Guidelines," RFC 3439, http://www.ietf.org/rfc/rfc3439.txt (updates RFC 1958), December 2002.

[I-D.ietf-cdni-problem-statement] Niven-Jenkins, B., Faucheur, F.,

and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-00 (work progress), September 2011.

[I-D.ietf-cdni-requirements] Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-00 (work in progress), September 2011.

[I-D.ietf-cdni-use-cases] Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Ma, K., "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

5.2 Informative reference

[I-D.stiemerling-cdni-routing-cons] Stiemerling, M., "Considerations on Request Routing for CDNI", draft-stiemerling-cdni-routing-cons-00, July 2011

[Stockhammer2011] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles", ACM Proceedings of the second annual ACM conference on Multimedia systems, 2011.

[3GP-DASH] ETSI TS 126 247 v10.0.0 (2011-06) Universal Mobile Telecommunications System (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (3GPP TS 26.247 version 10.0.0 Release 10).

Authors' Addresses

Yannick Le Louedec
Orange Labs, France
Email: yannick.lelouedec@orange.com

Christian Timmerer
UNI-KLU, Austria
Email: christian.timmerer@itec.uni-klu.ac.at

Spiros Spirou
Intracom, Greece
Email: spis@intracom.com

David Griffin
UCL, UK
Email: dgriffin@ee.ucl.ac.uk

Theodore Zahariadis
Synelixis, Greece
Email: zahariad@synelixis.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 3, 2012

B. Niven-Jenkins
Velocix (Alcatel-Lucent)
F. Le Faucheur
Cisco
N. Bitar
Verizon

October 31, 2011

Content Distribution Network Interconnection (CDNI) Problem Statement
draft-ietf-cdni-problem-statement-01

Abstract

Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for End Users and increased robustness of delivery. For these reasons they are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. It is generally desirable that a given content item can be delivered to an End User regardless of that End User's location or attachment network. This creates a requirement for interconnecting standalone CDNs so they can interoperate as an open content delivery infrastructure for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users. However, no standards or open specifications currently exist to facilitate such CDN interconnection.

The goal of this document is to outline the problem area of CDN interconnection (CDNI) for the IETF.

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 May 3, 2012.

Table of Contents

1. Introduction

The volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future. In the face of this growth, Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for End Users and increased robustness of delivery. For these reasons CDNs are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs.

It is generally desirable that a given content item can be delivered to an End User regardless of that End User's location or attachment network. However, the footprint of a given CDN in charge of delivering a given content may not expand close enough to the End User's current location or attachment network to realize the cost benefit and user experience that a more distributed CDN would provide. This creates a requirement for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users. For example, a CSP could contract with an "authoritative" CDN for the delivery of content and that authoritative CDN could contract with one or more downstream CDN(s) to distribute and deliver some or all of the content on behalf of the authoritative CDN. The formation and details of any business relationships between a CSP and a CDN and between one CDN and another CDN are out of scope of this document. However, no standards or open specifications currently exist to facilitate such CDN interconnection.

The goal of this document is to outline the problem area of CDN interconnection (CDNI) for the IETF. Section 2 discusses the use cases for CDN interconnection. Section 3 presents the CDNI model and problem area being considered by the IETF. Section 4 describes each CDNI interface individually and highlights example candidate protocols that could considered for reuse or leveraging to implement the CDNI interfaces. Section 5 provides a gap analysis against the work of other standards organizations. Section 6 describes the relationships between the CDNI problem space and other relevant IETF Working Groups.

1.1. Terminology

This document uses the following terms:

Content: Any form of digital data. One important form of Content with additional constraints on Distribution and Delivery is

continuous media (i.e. where there is a timing relationship between source and sink).

Metadata: Metadata in general is data about data.

Content Metadata: This is metadata about Content. Content Metadata comprises:

1. Metadata that is relevant to the distribution of the content (and therefore relevant to a CDN involved in the delivery of that content). We refer to this type of metadata as "Content Distribution Metadata". See also the definition of Content Distribution Metadata.
2. Metadata that is associated with the actual Content (and not directly relevant to the distribution of that Content) or content representation. For example, such metadata may include information pertaining to the Content's genre, cast, rating, etc as well as information pertaining to the Content representation's resolution, aspect ratio, etc.

Content Distribution Metadata: The subset of Content Metadata that is relevant to the distribution of the content. This is the metadata required by a CDN in order to enable and control content distribution and delivery by the CDN. In a CDN Interconnection environment, some of the Content Distribution Metadata may have an intra-CDN scope (and therefore need not be communicated between CDNs), while some of the Content Distribution Metadata have an inter-CDN scope (and therefore needs to be communicated between CDNs).

CDNI Metadata: Content Distribution Metadata with inter-CDN scope. For example, CDNI Metadata may include geo-blocking information (i.e. information defining geographical areas where the content is to be made available or blocked), availability windows (i.e. information defining time windows during which the content is to be made available or blocked) and access control mechanisms to be enforced (e.g. URI signature validation). CDNI Metadata may also include information about desired distribution policy (e.g. prepositioned vs dynamic acquisition) and about where/how a CDN can acquire the content. CDNI Metadata may also include content management information (e.g. request for deletion of Content from Surrogates) across interconnected CDNs.

Dynamic content acquisition: Dynamic content acquisition is where a CDN acquires content from the content source in response to an End User requesting that content from the CDN. In the context of CDN Interconnection, dynamic acquisition means that a downstream CDN does not acquire the content from content sources (including upstream CDNs) until a request for that content has been delegated to the

downstream CDN by an Upstream CDN.

Dynamic CDNI metadata acquisition: In the context of CDN Interconnection, dynamic CDNI metadata acquisition means that a downstream CDN does not acquire CDNI metadata for content from the upstream CDN until a request for that content has been delegated to the downstream CDN by an Upstream CDN.

Pre-positioned content acquisition: Content Pre-positioning is where a CDN acquires content from the content source prior to or independent of any End User requesting that content from the CDN. In the context of CDN interconnection the Upstream CDN instructs the Downstream CDN to acquire the content from content sources (including upstream CDNs) in advance of or independent of any End User requesting it.

Pre-positioned CDNI Metadata acquisition: In the context of CDN Interconnection, CDNI Metadata pre-positioning is where the Downstream CDN acquires CDNI metadata for content prior to or independent of any End User requesting that content from the Downstream CDN.

End User (EU): The 'real' user of the system, typically a human but maybe some combination of hardware and/or software emulating a human (e.g. for automated quality monitoring etc.)

User Agent (UA): Software (or a combination of hardware and software) through which the End User interacts with a Content Service. The User Agent will communicate with a Content Service for the selection of content and one or more CDNs for the delivery of the Content. Such communication is not restricted to HTTP and may be via a variety of protocols. Examples of User Agents (non-exhaustive) are: Browsers, Set Top Boxes (STBs), dedicated content applications (e.g. media players), etc.

Network Service Provider (NSP): Provides network-based connectivity/ services to End Users.

Content Service Provider (CSP): Provides a Content Service to End Users (which they access via a User Agent). A CSP may own the Content made available as part of the Content Service, or may license content rights from another party.

Content Service: The service offered by a Content Service Provider. The Content Service encompasses the complete service which may be wider than just the delivery of items of Content, e.g. the Content Service also includes any middleware, key distribution, program guide, etc. which may not require any direct interaction with the

CDN.

Content Distribution Network (CDN) / Content Delivery Network (CDN): Network infrastructure in which the network elements cooperate at layers 4 through layer 7 for more effective delivery of Content to User Agents. Typically a CDN consists of a Request Routing system, a Distribution System (that includes a set of Surrogates), a Logging System and a CDN control system.

CDN Provider: The service provider who operates a CDN. Note that a given entity may operate in more than one role. For example, a company may simultaneously operate as a Content Service Provider, a Network Service Provider and a CDN Provider.

CDN Interconnection (CDNI): The set of interfaces over which two or more CDNs communicate with each other in order to achieve the delivery of content to User Agents by Surrogates in one CDN (the downstream CDN) on behalf of another CDN (the upstream CDN).

Authoritative CDN: A CDN which has a direct relationship with a CSP for the distribution & delivery of that CSP’s content.

Upstream CDN: For a given End User request, the CDN (within a pair of directly interconnected CDNs) that redirects the request to the other CDN.

Downstream CDN: For a given End User request, the CDN (within a pair of directly interconnected CDNs) to which the request is redirected by the other CDN (the Upstream CDN). Note that in the case of successive redirections (e.g. CDN1-->CDN2-->CDN3) a given CDN (e.g. CDN2) may act as the Downstream CDN for a redirection (e.g. CDN1-->CDN2) and as the Upstream CDN for the subsequent redirection of the same request (e.g. CDN2-->CDN3).

Over-the-top (OTT): A service, e.g. a CDN, operated by a different operator than the NSP to which the users of that service are attached.

Surrogate: A device/function that interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content.

Request Routing System: The function within a CDN responsible for receiving a content request from a User Agent, obtaining and maintaining necessary information about a set of candidate surrogates or candidate CDNs, and for selecting and redirecting the user to the appropriate surrogate or CDN. To enable CDN Interconnection, the Request Routing System must also be capable of handling User Agent

content requests passed to it by another CDN.

Distribution System: The function within a CDN responsible for distributing Content Distribution Metadata as well as the Content itself inside the CDN (e.g. down to the surrogates).

Delivery: The function within CDN surrogates responsible for delivering a piece of content to the User Agent. For example, delivery may be based on HTTP progressive download or HTTP adaptive streaming.

Logging System: The function within a CDN responsible for collecting the measurement and recording of distribution and delivery activities. The information recorded by the logging system may be used for various purposes including charging (e.g. of the CSP), analytics and monitoring.

1.2. CDN Background

Readers are assumed to be familiar with the architecture, features and operation of CDNs. For readers less familiar with the operation of CDNs, the following resources may be useful:

- RFC 3040 [RFC3040] describes many of the component technologies that are used in the construction of a CDN.
- Taxonomy [TAXONOMY] compares the architecture of a number of CDNs.
- RFC 3466 [RFC3466] and RFC 3570 [RFC3570] are the output of the IETF Content Delivery Internetworking (CDI) working group which was closed in 2003.

Note: Some of the terms used in this document are similar to terms used the above referenced documents. When reading this document terms should be interpreted as having the definitions provided in Section 1.1.

2. CDN Interconnect Use Cases

An increasing number of NSPs are deploying CDNs in order to deal cost-effectively with the growing usage of on-demand video services and other content delivery applications.

CDNs allow caching of content closer to the edge of a network so that a given item of content can be delivered by a CDN Surrogate (i.e. a cache) to multiple User Agents (and their End Users) without transiting multiple times through the network core (i.e from the content origin to the surrogate). This contributes to bandwidth cost reductions for the NSP and to improved quality of experience for the

End Users. CDNs also enable replication of popular content across many surrogates, which enables content to be served to large numbers of User Agents concurrently. This also helps dealing with situations such as flash crowds and denial of service attacks.

The CDNs deployed by NSPs are not just restricted to the delivery of content to support the Network Service Provider's own 'walled garden' services, such as IP delivery of television services to Set Top Boxes, but are also used for delivery of content to other devices including PCs, tablets, mobile phones etc.

Some service providers operate over multiple geographies and federate multiple affiliate NSPs. These NSPs typically operate independent CDNs. As they evolve their services (e.g. for seamless support of content services to nomadic users across affiliate NSPs) there is a need for interconnection of these CDNs. However there are no open specifications, nor common best practices, defining how to achieve such CDN interconnection.

CSPs have a desire to be able to get (some of) their content to very large number of End Users and/or over many/all geographies and/or with a high quality of experience, all without having to maintain direct business relationships with many different CDN providers (or having to extend their own CDN to a large number of locations). Some NSPs are considering interconnecting their respective CDNs (as well as possibly over-the-top CDNs) so that this collective infrastructure can address the requirements of CSPs in a cost effective manner. In particular, this would enable the CSPs to benefit from on-net delivery (i.e. within the Network Service Provider's own network/CDN footprint) whenever possible and off-net delivery otherwise, without requiring the CSPs to maintain direct business relationships with all the CDNs involved in the delivery. Again, for this requirement, CDN providers (NSPs or over-the-top CDN operators) are faced with a lack of open specifications and best practices.

NSPs have often deployed CDNs as specialized cost-reduction projects within the context of a particular service or environment, some NSPs operate separate CDNs for separate services. For example, there may be a CDN for managed IPTV service delivery, a CDN for web-TV delivery and a CDN for video delivery to Mobile terminals. As NSPs integrate their service portfolio, there is a need for interconnecting these CDNs. Again, NSPs face the problem of lack of open interfaces for CDN interconnection.

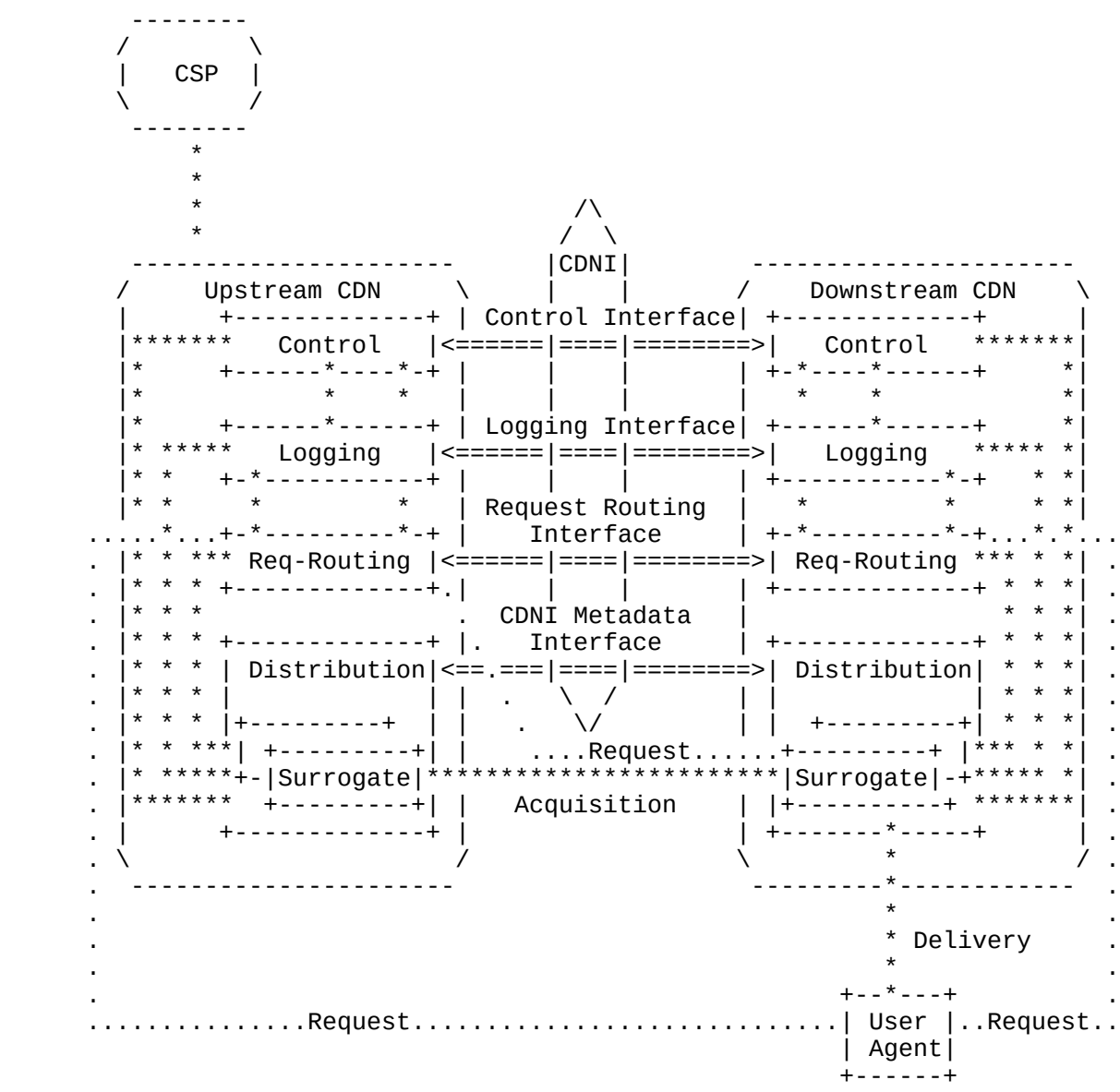
For operational reasons (e.g. disaster, flash crowd) or commercial reasons, an over-the-top CDN may elect to make use of another CDN (e.g. an NSP CDN with on-net Surrogates for a given footprint) for serving a subset of the user requests (e.g. requests from users

attached to that NSP). Again, for this requirement, CDN providers (over-the-top CDN providers or NSPs) are faced with a lack of open specifications and best practices.

Use cases for CDN Interconnection are further discussed in [I-D.ietf-cdni-use-cases].

3. CDN Interconnect Model & Problem Area for IETF

Interconnecting CDNs involves interactions among multiple different functions and components that form each CDN. Only some of those require standardization. This section discusses the problem area for the IETF work on CDN Interconnection. The CDNI model and problem area defined for IETF work is illustrated in Figure 1.



<==> interfaces inside the scope of CDNI
**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Problem Area

Listed below are the four interfaces required to interconnect a pair

of CDNs and that constitute the problem space that is proposed to be addressed by a potential CDNI working group in the IETF. The use of the term "interface" is meant to encompass the protocol over which CDNI data representations (e.g. CDNI Metadata records) are exchanged as well as the specification of the data representations themselves (i.e. what properties/fields each record contains, its structure, etc.).

- o CDNI Control interface: This interface allows the "CDNI Control" system in interconnected CDNs to communicate. This interface may support the following:
  - * Allow bootstrapping of the other CDNI interfaces (e.g. interface address/URL discovery and establishment of security associations).
  - * Allow configuration of the other CDNI interfaces (e.g. Upstream CDN specifies information to be reported through the CDNI Logging interface).
  - * Allow the downstream CDN to communicate static (or fairly static) information about its delivery capabilities and policies.
  - * Allow bootstrapping of the interface between CDNs for content acquisition (even if that interface itself is outside the scope of the CDNI work).
  - * Allow upstream CDN to initiate or request specific actions to be undertaken in the downstream CDN. For example, this may include the following capabilities:
    - + Allow an upstream CDN to request that content files and/or CDNI Metadata that it previously shared, be purged from, or invalidated in, a downstream CDN. Support for content deletion or invalidation from a CDN is a key requirement for some Content Service Providers in order, amongst other use cases for content deletion, to support the content rights agreements they have negotiated. Today’s CDNs use proprietary control interfaces to enable CSPs to remove content cached in the CDN and therefore there is a need to have a similar but standardized content deletion capability between interconnected CDNs.
    - + Allow an upstream CDN to initiate Pre-positioned content acquisition and/or Pre-positioned CDNI Metadata acquisition in a downstream CDN.
- o CDNI Request Routing interface: This interface allows the Request Routing systems in interconnected CDNs to communicate to ensure that an End User request can be (re)directed from an upstream CDN to a surrogate in the downstream CDN, in particular where selection responsibilities may be split across CDNs (for example the upstream CDN may be responsible for selecting the downstream CDN while the downstream CDN may be responsible for selecting the actual surrogate within that downstream CDN). In particular, the

CDN Request Routing interface, may support the following:
* Allow the upstream CDN to query the downstream CDN at request routing time before redirecting the request to the downstream CDN.
* Allow the downstream CDN to provide to the upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the downstream CDN by the upstream CDN request routing system when processing subsequent content requests from User Agents.

- o CDNI Metadata distribution interface: This interface allows the Distribution system in interconnected CDNs to communicate to ensure CDNI Metadata can be exchanged across CDNs. See Section 1.1 for definition and examples of CDNI Metadata.
- o CDNI Logging interface: This interface allows the Logging system in interconnected CDNs to communicate the relevant activity logs in order to allow log consuming applications to operate in a multi-CDN environments. For example, an upstream CDN may collect delivery logs from a downstream CDN in order to perform consolidated charging of the CSP or for settlement purposes across CDNs. Similarly, an upstream CDN may collect delivery logs from a downstream CDN in order to provide consolidated reporting and monitoring to the CSP.

Note that the actual grouping of functionalities under these four interfaces is considered tentative at this stage and may be changed after further study (e.g. some subset of functionality be moved from one interface into another).

The above list covers a significant potential problem space, in part because in order to interconnect two CDNs there are several 'touch points' that require standardization. However, it is expected that the CDNI interfaces need not be defined from scratch and instead can very significantly reuse or leverage existing protocols: this is discussed further in Section 4. Also, it is expected that the items above will be prioritized so that the CDNI Working Group can focus (at least initially) on the most essential and urgent work.

As part of the development of the CDNI interfaces and solutions it will also be necessary to agree on common mechanisms for how to identify and name the data objects that are to be interchanged between interconnected CDNs, as well as how to describe which policy should be used when doing so.

Some NSPs have started to perform experiments to explore whether their CDN use cases can already be addressed with existing CDN implementations. One set of such experiments is documented in [I-D.bertrand-cdni-experiments]. The conclusions of those experiments are that while some basic limited CDN Interconnection

functionality can be achieved with existing CDN technology, the current lack of any standardized CDNI interfaces/protocols such as those discussed in this document is preventing the deployment of production CDN Interconnection solutions with the necessary level of functionality.

The acquisition of content between interconnected CDNs is out of scope for CDNI and deserves some additional explanation. The consequence of such a decision is that the CDNI WG is focussed on only defining the control plane for CDNI; and the CDNI data plane (i.e. the acquisition & distribution of the actual content objects) will not be addressed by the CDNI WG. The rationale for such a decision is that CDNs today typically already use standardized protocols such as HTTP, FTP, rsync, etc. to acquire content from their CSP customers and it is expected that the same protocols could be used for acquisition between interconnected CDNs. Therefore the problem of content acquisition is considered already solved and all that is required from specifications developed by the CDNI WG is to describe within the CDNI Metadata where to go and which protocol to use to retrieve the content.

4. Design Approach for Realizing the CDNI APIs

This section expands on how CDNI interfaces can reuse and leverage existing protocols before describing each CDNI interface individually and highlighting example candidate protocols that could considered for reuse or leveraging to implement the CDNI interfaces. This discussion is not intended to pre-empt any WG decision as to the most appropriate protocols, technologies and solutions to select to solve CDNI but is intended as an illustration of the fact that the CDNI interfaces need not be created in a vacuum and that reuse or leverage of existing protocols is likely possible.

The four CDNI interfaces (CDNI Control interface, CDNI Request Routing interface, CDNI Metadata interface, CDNI Logging interface) described in Section 3 within the CDNI problem area are all control plane interfaces operating at the application layer (Layer 7 in the OSI network model). Since it is not expected that these interfaces would exhibit unique session, transport or network requirements as compared to the many other existing applications in the Internet, it is expected that the CDNI interfaces will be defined on top of existing session, transport and network protocols.

Although a new application protocol could be designed specifically for CDNI we assume that this is unnecessary and it is recommended that existing application protocols be reused or leveraged (HTTP [RFC2616], Atom Publishing Protocol [RFC5023], XMPP [RFC6120], for

example) to realize the CDNI interfaces.

4.1. CDNI Request Routing Interface

The CDNI Request Routing interface enables a Request Routing function in an upstream CDN to query a Request Routing function in a downstream CDN to determine if the downstream CDN is able (and willing) to accept the delegated content request and to allow the downstream CDN to control what the upstream Request Routing function should return to the User Agent in the redirection message.

The CDNI Request Routing interface needs to offer a mechanism for an upstream CDN to issue a "Redirection Request" to a downstream CDN. The Request Routing interface needs to be able to support scenarios where the initial User Agent request to the upstream CDN is received over DNS as well as over a content specific application protocol (e.g. HTTP, RTSP, RTMP, etc.).

Therefore a Redirection Request needs to contain information such as:

- The protocol (e.g. DNS, HTTP) over which the upstream CDN received the initial User Agent request.
- Additional details of the User Agent request that are required to perform effective Request Routing by the Downstream CDN. For DNS this would typically be the IP address of the DNS resolver making the request on behalf of the User Agent. For requests received over content specific application protocols the Redirection Request could contain significantly more information related to the original User Agent request but at a minimum would need to contain the User Agent's IP address, the equivalent of the HTTP Host header and the equivalent of the HTTP abs_path defined in [RFC2616].

It should be noted that, the CDNI architecture needs to consider that a downstream CDN may receive requests from User Agents without first receiving a Redirection Request from an upstream CDN, for example because:

- User Agents (or DNS resolvers) may cache DNS or application responses from Request Routers.
- Responses to Redirection Requests over the Request Routing interface may be cacheable.
- Some CDNs may want broader policies, e.g. CDN B agrees to always take CDN A's delegated redirection requests, in which case the necessary redirection details are exchanged out of band (of the CDNI interfaces), e.g. configured.

On receiving a Redirection Request, the downstream CDN will use the

information provided in the request to determine if it is able (and willing) to accept the delegated content request and needs to return the result of its decision to the upstream CDN.

Thus, a Redirection Response from the downstream CDN needs to contain information such as:

- Status code indicating acceptance or rejection (possibly with accompanying reasons).
- Information to allow redirection by the Upstream CDN. In the case of DNS-based request routing, this is expected to include the equivalent of a DNS record(s) (e.g. a CNAME) that the upstream CDN should return to the requesting DNS resolver. In the case of application based request routing, this is expected to include the application specific redirection response(s) to return to the requesting User Agent. For HTTP requests from User Agents this could be in the form of a URI that the upstream CDN could return in a HTTP 302 response.

The CDNI Request Routing interface is therefore a fairly straightforward request/response interface and could be implemented over any number of request/response protocols. For example, it may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.). This removes the need for the CDNI WG to define a new protocol for the request/response element of the CDNI Request Routing interface. Thus, the CDNI WG would be left only with the task of specifying:

- The recommended request/response protocol to use along with any additional semantics and procedures that are specific to the CDNI Request Routing interface (e.g. handling of malformed requests/ responses).
- The syntax (i.e representation/encoding) of the redirection requests and responses.
- The semantics (i.e. meaning and expected contents) of the redirection requests and responses.

Additionally, as discussed in Section 3, the CDNI Request Routing interface is also expected to enable a downstream CDN to provide to the upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the downstream CDN by the upstream CDN request routing system when processing subsequent content requests from User Agents. It is expected that such functionality of the CDNI request Routing could be specified by the CDNI WG with significant leveraging of existing IETF protocols supporting the dynamic distribution of reachability information (for example by leveraging existing routing protocols) or supporting application level queries for topological information (for example by

leveraging ALTO).

4.2. CDNI Metadata Interface

The CDNI Metadata interface enables the Metadata function in a downstream CDN to obtain CDNI Metadata from an upstream CDN so that the downstream CDN can properly process and respond to:

- Redirection Requests received over the CDNI Request Routing interface.
- Content Requests received directly from User Agents.

The CDNI Metadata interface needs to offer a mechanism for an Upstream CDN to:

- Distribute/update/remove CDNI Metadata to a Downstream CDN.

and/or to allow a downstream CDN to:

- Make direct requests for CDNI Metadata records where the downstream CDN knows the identity of the Metadata record(s) it requires.
- Search for CDNI Metadata records where the downstream CDN does not know the specific Metadata record(s) it requires but does know some property of the record it is searching for. For example, it may know the value of the HTTP Host header received in a HTTP request and it wants to obtain the CDNI Metadata for that host so that it can determine how to further process the received HTTP request.

The CDNI Metadata interface is therefore similar to the CDNI Request Routing interface because it is a request/response interface with the potential addition that CDNI Metadata search may have more complex semantics than a straightforward Request Routing redirection request. Therefore, like the CDNI Request Routing interface, the CDNI Metadata interface may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.) or possibly using other existing protocols such as XMPP [RFC6120]. This removes the need for the CDNI WG to define a new protocol for the request/response element of the CDNI Metadata interface.

Thus, the CDNI WG would be left only with the task of specifying:

- The recommended request/response protocol to use along with any additional semantics that are specific to the CDNI Metadata interface (e.g. handling of malformed requests/responses).
- The syntax (i.e representation/encoding) of the CDNI Metadata records that will be exchanged over the interface.

- The semantics (i.e. meaning and expected contents) of the individual properties of a Metadata record.
- How the relationships between different CDNI Metadata records are represented.

4.3. CDNI Logging Interface

The CDNI Logging interface enables details of logs or events to be exchanged between interconnected CDNs, where events could be:

- Log lines related to the delivery of content (similar to the log lines recorded in a web server's access log).
- Real-time or near-real time events before, during or after content delivery, e.g. content Start/Pause/Stop events, etc.
- Operations and diagnostic messages.

Within CDNs today, logs and events are used for a variety of purposes in addition to real-time and non real-time diagnostics and auditing by the CDN Provider and its customers. Specifically CDNs use logs to generate Call Data Records (CDRs) for passing to billing and payment systems and to real-time (and near real-time) analytics systems. Such use cases place requirements on the CDNI Logging interface to support guaranteed and timely delivery of log messages between interconnected CDNs. It may also be necessary to be able to prove the integrity of received log messages.

Several protocols already exist that could potentially be used to exchange CDNI logs between interconnected CDNs including SNMP Traps, syslog, ftp, HTTP POST, etc. although it is likely that some of the candidate protocols may not be well suited to meet all the requirements of CDNI. For example SNMP traps pose scalability concerns and SNMP does not support guaranteed delivery of Traps and therefore could result in log records being lost and the consequent CDRs and billing records for that content delivery not being produced as well as that content delivery being invisible to any analytics platforms.

Although it is not necessary to define a new protocol for exchanging logs across the CDNI Logging interface, the CDNI WG would still need to specify:

- The recommended protocol to use.
- A default set of log fields and their syntax & semantics. Today there is no standard set of common log fields across different content delivery protocols and in some cases there is not even a standard set of log field names and values for different implementations of the same delivery protocol.

- o A default set of events that trigger logs to be generated.

4.4. CDNI Control Interface

The CDNI Control interface allows the "CDNI Control" system in interconnected CDNs to communicate. The exact inter-CDN control functionality required to be supported by the CDNI Control interface is less well defined than the other three CDNI interfaces at this time.

However, as discussed in Section 3, the CDNI Control interface may be required to support functionality similar to the following:

- o Allow an upstream CDN and downstream CDN to establish, update or terminate their CDNI interconnection.
- o Allow bootstrapping of the other CDNI interfaces (e.g. protocol address discovery and establishment of security associations).
- o Allow configuration of the other CDNI interfaces (e.g. Upstream CDN specifies information to be reported through the CDNI Logging interface).
- o Allow the downstream CDN to communicate static information about its delivery capabilities, resources and policies.
- o Allow bootstrapping of the interface between CDNs for content acquisition (even if that interface itself is outside the scope of the CDNI work).

It is expected that for the Control interface also, existing protocols can be reused or leveraged. Those will be considered once the requirements for the Control interface have been refined.

5. Gap Analysis of relevant Standardization Activities

There are a number of other standards bodies and industry forums that are working in areas related to CDNs, and in some cases related to CDNI. This section outlines any potential overlap with the work of the CDNI WG and any component that could potentially be reused by CDNI.

A number of standards bodies have produced specifications related to CDNs, for example:

- o TISPAN has a dedicated specification for CDN.
- o OIPF and ATIS specify the architecture and the protocols of an IPTV solution. Although OIPF and ATIS specifications include the interaction with a CDN, the CDN specifications are coupled with their IPTV specifications.
- o CableLabs, SNIA and ITU have defined (or are working on) definitions for content related metadata definitions and specification for its distribution. However, they do not include

metadata specific to the distribution of content within a CDN or between interconnected CDNs.

- IETF CDI WG (now concluded) touched on the same problem space as the present document. However, in accordance with its initial charter, the CDI WG did not define any protocols or interfaces to actually enable CDN Interconnection and at that time (2003) there was not enough industry interest and real life requirements to justify rechartering the WG to conduct the corresponding protocol work.

Although some of the specifications describe multi-CDN cooperation or include reference points for interconnecting CDNs, none of them specify in sufficient detail all the CDNI interfaces and CDNI Metadata representations required to enable even a base level of CDN Interconnection functionality to be implemented.

The following sections will summarize the existing work of the standard bodies listed earlier against the CDNI problem space. Section 5.1 summarises existing interfaces that could be leveraged for content acquisition between CDNs and Section 5.2 summarises existing metadata specifications that may be applicable to CDNI. To date we are not aware of any standardisation activities in the areas of the remaining CDNI interfaces (CDNI Request Routing, CDNI Control and CDNI Logging).

5.1. Content Acquisition across CDNs and Delivery to End User (Data plane)

A number of standards bodies have completed work in the areas of content acquisition interface between a CSP and a CDN, as well as as on the delivery interface between the surrogate and the User Agent. Some of this work is summarized below.

TISPAN, OIPF and ATIS have specified IPTV and/or CoD services, including the data plane aspects (typically different flavors of RTP/RTCP and HTTP) to obtain content and deliver it to User Agents. For example, :

- The OIPF data plane includes both RTP and HTTP flavors (HTTP progressive download, HTTP Adaptive streaming [_3GP-DASH]).
- The ATIS specification "IPTV Content on Demand (CoD) Service" [ATIS-COD] defines a reference point (C2) and the corresponding HTTP-based data plane protocol for content acquisition between an authoritative origin server and the CDN.

While these protocols have not been explicitly specified for content acquisition across CDNs, they are suitable (in addition to others such as standard HTTP) for content acquisition between CDNs in a CDN Interconnection environment. Therefore for the purpose of the CDNI WG there are already multiple existing data plane protocols that can

be used for content acquisition across CDNs.

Similarly, there are multiple existing standards (e.g. the OIPF data plane mentioned above, HTTP adaptive streaming [_3GP-DASH]) or public specifications (e.g. vendor specific HTTP Adaptive streaming specifications) so that content delivery can be considered already solved (or at least sufficiently addressed in other forums

Thus, specification of the content acquisition interface between CDNs and the delivery interface between the surrogate and the User Agent are out of scope for CDNI. CDNI may only concern itself with the negotiation/selection aspects of the acquisition protocol to be used in a CDN interonnect scenario.

5.2. CDNI Metadata

CableLabs, ITU, OIPF and TV-Anytime have work items dedicated to the specification of content metadata:

- CableLabs has defined specifications for CoD Content Metadata as part of its VOD Metadata project. "The VOD Metadata project is a cable television industry and cross-industry-wide effort to specify the metadata and interfaces for distribution of video-on-demand (VOD) material from multiple content providers to cable operators." [CableLabs-Metadata]. However, while the CableLabs work specifies an interface between a content provider and a service provider running a CDN, it does not include an interface that could be used between CDNs.
- ITU Study Group 16 has started work on a number of draft Recommendations (H.IPTV-CPMD, H.IPTV-CPMD, HSTP.IPTV-CMA, HSTP.IPTV-UMCI) specifying metadata for content distribution in IPTV services.
- An Open IPTV Terminal receives the technical description of the content distribution from the OIPF IPTV platform before receiving any content. The Content distribution metadata is sent in the format of a TV-Anytime XSD including tags to describes the location and program type (on demand or Live) as well as describing the time availability of the on demand and live content.

However the specifications outlined above do not include metadata specific to the distribution of content within a CDN or between interconnected CDNs, for example geo-blocking information, availability windows, access control mechanisms to be enforced by the surrogate, how to map an incoming content request to a file on the origin server or acquire it from the upstream CDN etc.

The CDMI standard ([SNIA-CDMI]) from SNIA defines metadata that can

be associated with data that is stored by a cloud storage provider. While the metadata currently defined do not match the need of a CDN Interconnection solution, it is worth considering CDMI as one of the existing pieces of work that may potentially be leveraged for the CDNI Metadata interface (e.g by extending the CDMI metadata to address more specific CDNI needs).

6. Relationship to relevant IETF Working Groups

6.1. ALTO

As stated in the ALTO Working Group charter [ALTO-Charter]:

"The Working Group will design and specify an Application-Layer Traffic Optimization (ALTO) service that will provide applications with information to perform better-than-random initial peer selection. ALTO services may take different approaches at balancing factors such as maximum bandwidth, minimum cross-domain traffic, lowest cost to the user, etc. The WG will consider the needs of BitTorrent, tracker-less P2P, and other applications, such as content delivery networks (CDN) and mirror selection."

In particular, the ALTO service can be used by a CDN Request Routing system to improve its selection of a CDN surrogate to serve a particular User Agent request (or to serve a request from another surrogate). [I-D.jenkins-alto-cdn-use-cases] describes a number of use cases for a CDN to be able to obtain network topology and cost information from an ALTO server(s) and [I-D.penno-alto-cdn] discusses how CDN Request Routing could be used as an integration point of ALTO into CDNs. It is possible that the ALTO service could be used in the same manner in a multi-CDN environment based on CDN Interconnection. For example, an upstream CDN may take advantage of the ALTO service in its decision for selecting a downstream CDN to which a user request should be delegated.

However, the work of ALTO is complementary to and does not overlap with the work described in this document because the integration between ALTO and a CDN is an internal decision for a specific CDN and is therefore out of scope for the CDNI WG. One area for further study is whether additional information should be provided by an ALTO service to facilitate CDNI CDN selection.

6.2. DECADE

The DECADE Working Group [DECADE-Charter] is addressing the problem of reducing traffic on the last-mile uplink, as well as backbone and transit links caused by P2P streaming and file sharing applications.

It addresses the problem by enabling an application endpoint to make content available from an in-network storage service and by enabling other application endpoints to retrieve the content from there.

Exchanging data through the in-network storage service in this manner, instead of through direct communication, provides significant gain where:

- The network capacity/bandwidth from in-network storage service to application endpoint significantly exceeds the capacity/bandwidth from application endpoint to application endpoint (e.g. because of an end-user uplink bottleneck); and
- Where the content is to be accessed by multiple instances of application endpoints (e.g. as is typically the case for P2P applications).

While, as is the case for any other data distribution application, the DECADE architecture and mechanisms could potentially be used for exchange of CDNI control plane information via an in-network-storage service (as opposed to directly between the entities terminating the CDNI interfaces in the neighbor CDNs), we observe that:

- CDNI would operate as a "Content Distribution Application" from the DECADE viewpoint (i.e. would operate on top of DECADE).
- There does not seem to be obvious benefits in integrating the DECADE control plane responsible for signaling information relating to control of the in-network storage service itself, and the CDNI control plane responsible for application-specific CDNI interactions (such as exchange of CDNI metadata, CDNI request redirection, transfer of CDNI logging information).
- There would typically be limited benefits in making use of a DECADE in-network storage service because the CDNI interfaces are expected to be terminated by a very small number of CDNI clients (if not one) in each CDN, and the CDNI clients are expected to benefit from high bandwidth/capacity when communicating directly to each other (at least as high as if they were communicating via an in-network storage server).

The DECADE in-network storage architecture and mechanisms may theoretically be used for the acquisition of the content objects themselves between interconnected CDNs. It is not expected that this would have obvious benefits in typical situations where a content object is acquired only once from an Upstream CDN to a Downstream CDN (and then distributed as needed inside the Downstream CDN). But it might have benefits in some particular situations. Since the acquisition protocol between CDNs is outside the scope of the CDNI work, this question is left for further study.

The DECADE in-network storage architecture and mechanisms may potentially also be used within a given CDN for the distribution of the content objects themselves among surrogates of that CDN. Since the CDNI work does not concern itself with operation within a CDN, this question is left for further study.

Therefore, the work of DECADE may be complementary to but does not overlap with the CDNI work described in this document.

6.3. PPSP

As stated in the PPSP Working Group charter [PPSP-Charter]:

"The Peer-to-Peer Streaming Protocol (PPSP) working group develops two signaling and control protocols for a peer-to-peer (P2P) streaming system for transmitting live and time-shifted media content with near real-time delivery requirements." and "The PPSP WG designs a protocol for signaling and control between trackers and peers (the PPSP "tracker protocol") and a signaling and control protocol for communication among the peers (the PPSP "peer protocol"). The two protocols enable peers to receive streaming data within the time constraints required by specific content items."

Therefore PPSP is concerned with the distribution of the streamed content itself along with the necessary signaling and control required to distribute the content. As such, it could potentially be used for the acquisition of streamed content across interconnected CDNs. But since the acquisition protocol is outside the scope of the work proposed for CDNI, we leave this for further study. Also, because of its streaming nature, PPSP is not seen as applicable to the distribution and control of the CDNI control plane and CDNI data representations.

Therefore, the work of PPSP may be complementary to but does not overlap with the work described in this document for CDNI.

7. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Security Considerations

Distribution of content by a CDN comes with a range of security

considerations such as how to enforce control of access to the content by users in line with the CSP policy. These security aspects are already dealt with by CDN Providers and CSPs today in the context of standalone CDNs. However, interconnection of CDNs introduces a new set of security considerations by extending the trust model (i.e. the CSP "trusts" a CDN that "trusts" another CDN).

Maintaining the security of the content itself, its associated metadata (including distribution and delivery policies) and the CDNs distributing and delivering it, are critical requirements for both CDN Providers and CSPs and any work on CDN Interconnection must provide sufficient mechanisms to maintain the security of the overall system of interconnected CDNs as well as the information (content, metadata, logs, etc) distributed and delivered through any CDN interconnects.

9. Acknowledgements

The authors would like to thank Andre Beck, Gilles Bertrand, Mark Carlson, Bruce Davie, David Ferguson, Yiu Lee, Kent Leung, Will Li, Kevin Ma, Julien Maisonneuve, Guy Meador, Emile Stephan, Oskar van Deventer and Mahesh Viveganandhan for their review comments and contributions to the text.

10. References

10.1. Normative References

10.2. Informative References

[ALTO-Charter] "IETF ALTO WG Charter (http://datatracker.ietf.org/wg/alto/charter/)".

[ATIS] "ATIS (http://www.atis.org/)".

[ATIS-COD] "ATIS IIF: IPTV Content on Demand Service, January 2011 (http://www.atis.org/iif/_Com/Docs/Task_Forces/ARCH/ATIS-0800042.pdf)".

[CDI-Charter] "IETF CDI WG Charter (http://www.ietf.org/wg/concluded/cdi)".

[CableLabs]

"CableLabs (http://www.cablelabs.com/about/)".

[CableLabs-Metadata]
"CableLabs VoD Metadata Project Primer (http://www.cablelabs.com/projects/metadata/primer/)".

[DECADE-Charter]
"IETF DECADE WG Charter (http://datatracker.ietf.org/wg/decade/charter/)".

[I-D.bertrand-cdni-experiments]
Bertrand, G., Faucheur, F., and L. Peterson, "Content Distribution Network Interconnection (CDNI) Experiments", draft-bertrand-cdni-experiments-01 (work in progress), August 2011.

[I-D.ietf-cdni-use-cases]
Bertrand, G., Emile, S., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

[I-D.jenkins-alto-cdn-use-cases]
Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and S. Previdi, "Use Cases for ALTO within CDNs", draft-jenkins-alto-cdn-use-cases-01 (work in progress), June 2011.

[I-D.penno-alto-cdn]
Penno, R., Medved, J., Alimi, R., Yang, R., and S. Previdi, "ALTO and Content Delivery Networks", draft-penno-alto-cdn-03 (work in progress), March 2011.

[MPEG-DASH]
"Information technology - MPEG systems technologies - Part 6: Dynamic adaptive streaming over HTTP (DASH), (DIS version), February 2011 http://mpeg.chiariglione.org/working_documents.htm#MPEG-B".

[OIPF-Overview]
"OIPF Release 2 Specification Volume 1 - Overview", September 2010.

[P2PRG-CDNI]
Davie, B. and F. Le Faucheur, "Interconnecting CDNs aka "Peering Peer-to-Peer" (http://www.ietf.org/proceedings/77/slides/P2PRG-2.pdf)",

March 2010.

[PPSP-Charter]
"IETF PPSP WG Charter
(http://datatracker.ietf.org/wg/ppsp/charter/)".

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3040] Cooper, I., Melve, I., and G. Tomlinson, "Internet Web Replication and Caching Taxonomy", RFC 3040, January 2001.

[RFC3466] Day, M., Cain, B., Tomlinson, G., and P. Rzewski, "A Model for Content Internetworking (CDI)", RFC 3466, February 2003.

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

[RFC3570] Rzewski, P., Day, M., and D. Gilletti, "Content Internetworking (CDI) Scenarios", RFC 3570, July 2003.

[RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[SNIA-CDMI]
"SNIA CDMI (http://www.snia.org/tech_activities/standards/curr_standards/cdmi)".

[TAXONOMY]
Pathan, A., "A Taxonomy and Survey of Content Delivery Networks
(http://www.gridbus.org/reports/CDN-Taxonomy.pdf)", 2007.

[Y.1910] "ITU-T Recomendation Y.1910 "IPTV functional architecture"", September 2008.

[Y.2019] "ITU-T Recomendation Y.2019 "Content delivery functional architecture in NGN"", September 2010.

[_3GP-DASH]
"Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming

over HTTP (3GP-DASH)
http://www.3gpp.org/ftp/Specs/html-info/26247.htm".

Appendix A. Additional Material

Note to RFC Editor: This appendix is to be removed on publication as an RFC.

A.1. Non-Goals for IETF

Listed below are aspects of content delivery that the authors propose be kept outside of the scope of a potential CDNI working group:

- The interface between Content Service Provider and the Authoritative CDN (i.e. the upstream CDN contracted by the CSP for delivery by this CDN or by its downstream CDNs).
- The delivery interface between the delivering CDN surrogate and the User Agent, such as streaming protocols.
- The request interface between the User Agent and the request-routing system of a given CDN. Existing IETF protocols (e.g. HTTP, RTSP, DNS) are commonly used by User Agents to request content from a CDN and by CDN request routing systems to redirect the User Agent requests. The CDNI working group need not define new protocols for this purpose. Note however, that the CDNI control plane interface may indirectly affect some of the information exchanged through the request interface (e.g. URI).
- The content acquisition interface between CDNs (i.e. the data plane interface for actual delivery of a piece of content from one CDN to the other). This is expected to use existing protocols such as HTTP or protocols defined in other forums for content acquisition between an origin server and a CDN (e.g. HTTP-based C2 reference point of ATIS IIF CoD). The CDN Interconnection solution may only concern itself with the agreement/negotiation aspects of which content acquisition protocol is to be used between two interconnected CDNs in view of facilitating interoperability.
- End User/User Agent Authentication. End User/User Agent authentication and authorization are the responsibility of the Content Service Provider.
- Content preparation, including encoding and transcoding. The CDNI architecture aims at allowing distribution across interconnected CDNs of content treated as opaque objects. Interpretation and processing of the objects, as well as optimized delivery of these objects by the surrogate to the End User are outside the scope of CDNI.
- Digital Rights Management (DRM). DRM is an end-to-end issue between a content protection system and the User Agent.

- Applications consuming CDNI logs (e.g. charging, analytics, reporting,...).
- Internal CDN interfaces & protocols (i.e. interfaces & protocols within one CDN).
- Scalability of individual CDNs. While scalability of the CDNI interfaces/approach is in scope, how an individual CDN scales is out of scope.
- Actual algorithms for selection of CDNs or Surrogates by Request Routing systems (however, some specific parameters required as input to these algorithms may be in scope when they need to be communicated across CDNs).
- Surrogate algorithms. For example caching algorithms and content acquisition methods are outside the scope of the CDNI work. Content management (e.g. Content Deletion) as it relates to CDNI content management policies, is in scope but the internal algorithms used by a cache to determine when to no longer cache an item of Content (in the absence of any specific metadata to the contrary) is out of scope.
- Element management interfaces.
- Commercial, business and legal aspects related to the interconnections of CDNs.

A.2. Related standardization activities

A.2.1. IETF CDI Working Group (Concluded)

The Content Distribution Internetworking (CDI) Working Group was formed in the IETF following a BoF in December 2000 and closed in mid 2003.

For convenience, here is an extract from the CDI WG charter [CDI-Charter]:

"

- The goal of this working group is to define protocols to allow the interoperation of separately-administered content networks.
- A content network is an architecture of network elements, arranged for efficient delivery of digital content. Such content includes, but is not limited to, web pages and images delivered via HTTP, and streaming or continuous media which are controlled by RTSP.
- The working group will first define requirements for three modes of content internetworking: interoperation of request-routing systems, interoperation of distribution systems, and interoperation of accounting systems. These requirements are intended to lead to a follow-on effort to define protocols for interoperation of these systems.

- In its initial form, the working group is not chartered to deliver those protocols [...]

"

Thus, the CDI WG touched on the same problem space as the present document.

The CDI WG published 3 Informational RFCs:

- RFC 3466 [RFC3466] - "A Model for Content Internetworking (CDI)".
- RFC 3568 [RFC3568] - "Known Content Network (CN) Request-Routing Mechanisms".
- RFC 3570 [RFC3570] - "Content Internetworking (CDI) Scenarios".

A.2.2. 3GPP

3GPP was the first organization that released a specification related to adaptive streaming over HTTP. 3GPP Release 9 specification on adaptive HTTP streaming was published in March 2010, and there have been some bug fixes on this specification since the publication. In addition, 3GPP is preparing an extended version for Release 10, which is scheduled to be published later in 2011. This release will include a number of clarifications, improvements and new features.

[_3GP-DASH] is defined as a general framework independent of the data encapsulation format. It has support for fast initial startup and seeking, adaptive bitrate switching, re-use of HTTP origin and cache servers, re-use of existing media playout engines, on-demand, live and time-shifted delivery. It specifies syntax and semantics of Media Presentation Description (MPD), format of segments and delivery protocol for segments. It does not specify content provisioning, client behavior or transport of MPD.

The content retrieved by a client using [_3GP-DASH] adaptive streaming could be obtained from a CDN but this is not discussed or specified in the 3GPP specifications as it is transparent to [_3GP-DASH] operations. Similarly, it is expected that [_3GP-DASH] can be used transparently from the CDNs as a delivery protocol (between the delivering CDN surrogate and the User Agent) in a CDN Interconnection environment. [_3GP-DASH] could also be a candidate for content acquisition between CDNs in a CDN Interconnection environment.

A.2.3. ISO MPEG

Within ISO MPEG, the Dynamic Adaptive Streaming over HTTP (DASH) ad-hoc group adopted the 3GPP Release 9 [_3GP-DASH] specification as a

starting point and has made some improvements and extensions. Similar to 3GPP SA4, the MPEG DASH ad-hoc group has been working on standardizing the manifest file and the delivery format. Additionally, the MPEG DASH ad-hoc group has also been working on the use of MPEG-2 Transport Streams as a media format, conversion from/to existing file formats, common encryption, and so on. The MPEG DASH specification could also be a candidate for delivery to the User Agent and for content acquisition between CDNs in a CDN Interconnection environment. The Draft International Standard (DIS) version [MPEG-DASH] is currently publicly available since early February 2011.

In the 95th MPEG meeting in January 2011, the DASH ad-hoc group decided to start a new evaluation experiment called "CDN-EE". The goals are to understand the requirements for MPEG DASH to better support CDN-based delivery, and to provide a guidelines document for CDN operators to better support MPEG DASH streaming services. The ongoing work is still very preliminary and does not currently target looking into CDN Interconnection use cases.

A.2.4. ATIS IIF

ATIS ([ATIS]) IIF is the IPTV Interoperability Forum (within ATIS) that develops requirements, standards, and specifications for IPTV.

ATIS IIF is developing the "IPTV Content on Demand (CoD) Service" specification. This includes use of a CDN (referred to in ATIS IIF CoD as the "Content Distribution and Delivery Functions") for support of a Content on Demand (CoD) Service as part of a broader IPTV service. However, this only covers the case of a managed IPTV service (in particular where the CDN is administered by the service provider) and does not cover the use, or interconnection, of multiple CDNs.

A.2.5. CableLabs

"Founded in 1988 by cable operating companies, Cable Television Laboratories, Inc. (CableLabs) is a non-profit research and development consortium that is dedicated to pursuing new cable telecommunications technologies and to helping its cable operator members integrate those technical advancements into their business objectives." [CableLabs]

CableLabs has defined specifications for CoD Content Metadata as part of its VOD Metadata project.

A.2.6. ETSI MCD

ETSI MCD (Media Content Distribution) is the ETSI technical committee "in charge of guiding and coordinating standardization work aiming at the successful overall development of multimedia systems (television and communication) responding to the present and future market requests on media content distribution".

MCD created a specific work item on interconnection of heterogeneous CDNs ("CDN Interconnection, use cases and requirements") in March 2010. MCD very recently created a working group to progress this work item. However, no protocol level work has yet started in MCD for CDN Interconnection.

A.2.7. ETSI TISPAN

ETSI TISPAN has published two sets of IPTV specifications, one of which is based on IMS. In addition, TISPAN is about to complete the specifications of a CDN architecture supporting delivery of various content services such as time-shifted TV and VoD to TISPAN devices (UEs) or regular PCs. The use cases allow for hierarchically and geographically distributed CDN scenarios, along with multi-CDN cooperation. As a result, the architecture contains reference points to support interconnection of other TISPAN CDNs. The protocol definition phase for the corresponding CDN architecture was kicked-off at the end of 2010. In line with its long history of leveraging IETF protocols, ETSI could potentially leverage CDNI interfaces developed in the IETF for their related protocol level work on interconnections of CDNs.

A.2.8. ITU-T

SG13 is developing standards related to the support of IPTV services (i.e.. multimedia services such as television/VoD/audio/text/ graphics/data delivered over IP-based managed networks).

ITU-T Recommendation Y.1910 [Y.1910] provides the description of the IPTV functional architecture. This architecture includes functions and interfaces for the distribution and delivery of content. This architecture is aligned with the ATIS IIF architecture.

Based upon ITU-T Rec. Y.1910, ITU-T Rec. Y.2019 [Y.2019] describes in more detail the content delivery functional architecture. This architecture allows CDN Interconnection: some interfaces (such as D3, D4) at the control level allow relationships between different CDNs, in the same domain or in different domains. Generic procedures are described, but the choice of the protocols is open.

A.2.9. Open IPTV Forum (OIPF)

The Open IPTV Forum has developed an end-to-end solution to allow any OIPF terminal to access enriched and personalized IPTV services either in a managed or a non-managed network[OIPF-Overview]. Some OIPF services (such as Network PVR) may be hosted in a CDN.

To that end, the Open IPTV Forum specification is made of 5 parts:

- Media Formats including HTTP Adaptive Streaming
- Content Metadata
- Protocols
- Terminal (Declarative or Procedural Application Environment)
- Authentication, Content Protection and Service Protection

A.2.10. TV-Anytime Forum

Version 1 of the TV-Anytime Forum specifications were published as ETSI TS 102 822-1 through ETSI TS 102 822-7 "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime")". It includes the specification of content metadata in XML schemas (ETSI TS 102 822-3) which define technical parameters for the description of CoD and Live contents. The specification is referenced by DVB and OIPF.

The TV-anytime Forum was closed in 2005.

A.2.11. SNIA

The Storage Networking Industry Association (SNIA) is an association of producers and consumers of storage networking products whose goal is to further storage networking technology and applications.

SNIA has published the Cloud Data Management Interface (CDMI) standard ([SNIA-CDMI]).

"The Cloud Data Management Interface defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface."

A.3. Related Research Projects

A.3.1. IRTF P2P Research Group

Some information on CDN interconnection motivations and technical issues were presented in the P2P RG at IETF 77. The presentation can be found in [P2PRG-CDNI].

A.3.2. OCEAN

OCEAN (http://www.ict-ocean.eu/) is an EU funded research project that started in February 2010 for 3 years. Some of its objectives are relevant to CDNI. It aims, among other things, at designing a new architectural framework for audiovisual content delivery over the Internet, defining public interfaces between its major building blocks in order to foster multi-vendor solutions and interconnection between Content Networks (the term "Content Networks" corresponds here to the definition introduced in [RFC3466], which encompasses CDNs).

OCEAN has not yet published any open specifications, nor common best practices, defining how to achieve such CDN interconnection.

A.3.3. Eurescom P1955

Eurescom P1955 was a 2010 research project involving a four European Network operators, which studied the interests and feasibility of interconnecting CDNs by firstly elaborating the main service models around CDN interconnection, as well as analyzing an adequate CDN interconnection technical architecture and framework, and finally by providing recommendations for telcos to implement CDN interconnection. The Eurescom P1955 project ended in July 2010.

The authors are not aware of material discussing CDN interconnection protocols or interfaces made publically available as a deliverable of this project.

Authors' Addresses

Ben Niven-Jenkins
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK

Email: ben@velocix.com

Francois Le Faucheur
Cisco Systems
Greenside, 400 Avenue de Roumanille
Sophia Antipolis 06410
France

Phone: +33 4 97 23 26 19
Email: flefauch@cisco.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
USA

Email: nabil.bitar@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 27, 2012

B. Niven-Jenkins
Velocix (Alcatel-Lucent)
F. Le Faucheur
Cisco
N. Bitar
Verizon

June 25, 2012

Content Distribution Network Interconnection (CDNI) Problem Statement
draft-ietf-cdni-problem-statement-08

Abstract

Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for End Users and increased robustness of delivery. For these reasons they are frequently used for large-scale content delivery. As a result, existing CDN Providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. It is generally desirable that a given content item can be delivered to an End User regardless of that End User's location or attachment network. This is the motivation for interconnecting standalone CDNs so they can interoperate as an open content delivery infrastructure for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users. However, no standards or open specifications currently exist to facilitate such CDN interconnection.

The goal of this document is to outline the problem area of CDN interconnection for the IETF CDNI (CDN Interconnection) working group.

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 December 27, 2012.

Copyright Notice

Table of Contents

1. Introduction

The volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future. In the face of this growth, Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for End Users (EUs) and increased robustness of delivery. For these reasons CDNs are frequently used for large-scale content delivery. As a result, existing CDN Providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs.

It is generally desirable that a given content item can be delivered to an EU regardless of that EU's location or the network they are attached to. However, a given CDN in charge of delivering a given content may not have a footprint that expands close enough to the EU's current location or attachment network, or may not have the necessary resources, to realize the user experience and cost benefit that a more distributed CDN infrastructure would allow. This is the motivation for interconnecting standalone CDNs so that their collective CDN footprint and resources can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to EUs. As an example, a CSP could contract with an "authoritative" CDN Provider for the delivery of content and that authoritative CDN Provider could contract with one or more downstream CDN Provider(s) to distribute and deliver some or all of the content on behalf of the authoritative CDN Provider.

A typical end to end content delivery scenario would then involve the following business arrangements:

- A business arrangement between the EU and his CSP, authorizing access by the EU to content items controlled by the CSP.
- A business arrangement between the CSP and an "authoritative" CDN Provider where the CSP authorizes the CDN Provider to perform the content delivery on behalf of the CSP.
- A business arrangement between the authoritative CDN Provider and another (or other) CDN(s) where the authoritative CDN may delegate the actual serving of some of the content delivery requests to the other CDN(s). A particular case, is where this other CDN Provider happens to also be the Network Service Provider providing network access to the EU, in which case there is also a separate and independent business relationship between the EU and the NSP for the corresponding network access.

The formation and details of any business relationships between a CSP and a CDN Provider as well as between one CDN Provider and another CDN Provider are out of scope of this document. However, this

document concerns itself with the fact that no standards or open specifications currently exist to facilitate such CDN interconnection from a technical perspective.

One possible flow for performing an end to end content delivery across a CDN Interconnect is described below:

- o The initial request from an EU's User Agent is first received by the authoritative (upstream) CDN, which is the CDN with a business arrangement with the CSP.
- o The authoritative (upstream) CDN may serve the request itself, or it may elect to use CDN Interconnect to redirect the request to a downstream CDN that is in a better position to do so (e.g. a downstream CDN that is "closer" to the EU).
- o The EU's User Agent will "follow" the redirect returned by the authoritative CDN and request the content from the downstream CDN. If required the downstream CDN will acquire the requested content from the authoritative (upstream) CDN, and if necessary the authoritative CDN will acquire the requested content from the Content Service Provider.

The goal of this document is to outline the problem area of CDN interconnection. Section 2 discusses the use cases for CDN interconnection. Section 3 presents the CDNI model and problem area being considered by the IETF. Section 4 describes each CDNI interface individually and highlights example candidate protocols that could be considered for reuse or leveraging to implement the CDNI interfaces. Appendix B.2 describes the relationships between the CDNI problem space and other relevant IETF Working Groups and IRTF Reserach Groups.

1.1. Terminology

This document uses the following terms:

Content: Any form of digital data. One important form of Content with additional constraints on distribution and delivery is continuous media (i.e. where there is a timing relationship between source and sink).

Metadata: Metadata in general is data about data.

Content Metadata: This is metadata about Content. Content Metadata comprises:

1. Metadata that is relevant to the distribution of the content (and therefore relevant to a CDN involved in the delivery of that content). We refer to this type of metadata as "Content

Distribution Metadata". See also the definition of Content Distribution Metadata.

2. Metadata that is associated with the actual Content or content representation, and not directly relevant to the distribution of that Content. For example, such metadata may include information pertaining to the Content's genre, cast, rating, etc as well as information pertaining to the Content representation's resolution, aspect ratio, etc.

Content Distribution Metadata: The subset of Content Metadata that is relevant to the distribution of the content. This is the metadata required by a CDN in order to enable and control content distribution and delivery by the CDN. In a CDN Interconnection environment, some of the Content Distribution Metadata may have an intra-CDN scope (and therefore need not be communicated between CDNs), while some of the Content Distribution Metadata may have an inter-CDN scope (and therefore needs to be communicated between CDNs).

CDNI Metadata: Content Distribution Metadata with inter-CDN scope. For example, CDNI Metadata may include geo-blocking information (i.e. information defining geographical areas where the content is to be made available or blocked), availability windows (i.e. information defining time windows during which the content is to be made available or blocked) and access control mechanisms to be enforced (e.g. URI signature validation). CDNI Metadata may also include information about desired distribution policy (e.g. prepositioned vs dynamic acquisition) and about where/how a CDN can acquire the content. CDNI Metadata may also include content management information (e.g. request for deletion of Content from Surrogates) across interconnected CDNs.

Dynamic content acquisition: Dynamic content acquisition is where a CDN acquires content from the content source in response to an End User requesting that content from the CDN. In the context of CDN Interconnection, dynamic acquisition means that a downstream CDN acquires the content from content sources (including upstream CDNs) at some point in time after a request for that content is delegated to the downstream CDN by an Upstream CDN (and that specific content is not yet available in the downstream CDN).

Dynamic CDNI metadata acquisition: In the context of CDN Interconnection, dynamic CDNI metadata acquisition means that a downstream CDN acquires CDNI metadata for content from the upstream CDN at some point in time after a request for that content is delegated to the downstream CDN by an Upstream CDN (and that specific CDNI metadata is not yet available in the downstream CDN). See also the definitions for downstream CDN and upstream CDN.

Pre-positioned content acquisition: Content Pre-positioning is where a CDN acquires content from the content source prior to, or independently of, any End User requesting that content from the CDN. In the context of CDN interconnection the Upstream CDN instructs the Downstream CDN to acquire the content from content sources (including upstream CDNs) in advance of or independent of any End User requesting it.

Pre-positioned CDNI Metadata acquisition: In the context of CDN Interconnection, CDNI Metadata pre-positioning is where the Downstream CDN acquires CDNI metadata for content prior to or independent of any End User requesting that content from the Downstream CDN.

End User (EU): The 'real' user of the system, typically a human but maybe some combination of hardware and/or software emulating a human (e.g. for automated quality monitoring etc.)

User Agent (UA): Software (or a combination of hardware and software) through which the End User interacts with a Content Service. The User Agent will communicate with a Content Service for the selection of content and one or more CDNs for the delivery of the Content. Such communication is not restricted to HTTP and may be via a variety of protocols. Examples of User Agents (non-exhaustive) are: Browsers, Set Top Boxes (STBs), dedicated content applications (e.g. media players), etc.

Network Service Provider (NSP): Provides network-based connectivity/ services to End Users.

Content Service Provider (CSP): Provides a Content Service to End Users (which they access via a User Agent). A CSP may own the Content made available as part of the Content Service, or may license content rights from another party.

Content Service: The service offered by a Content Service Provider. The Content Service encompasses the complete service which may be wider than just providing access to items of Content, e.g. the Content Service also includes any middleware, key distribution, program guide, etc. which may not require any direct interaction with the CDN, or CDNs, involved in the distribution and delivery of the content.

Content Distribution Network (CDN) / Content Delivery Network (CDN): Network infrastructure in which the network elements cooperate at layers 4 through layer 7 for more effective delivery of Content to User Agents. Typically a CDN consists of a Request Routing system, a Distribution System (that includes a set of Surrogates), a Logging

System and a CDN control system.

CDN Provider: The service provider who operates a CDN and offers a service of content delivery, typically used by a Content Service Provider or another CDN Provider. Note that a given entity may operate in more than one role. For example, a company may simultaneously operate as a Content Service Provider, a Network Service Provider and a CDN Provider.

CDN Interconnection (CDNI): A relationship between a pair of CDNs that enables one CDN to provide content delivery services on behalf of another CDN. A CDN Interconnection may be wholly or partially realized through a set of interfaces over which a pair of CDNs communicate with each other in order to achieve the delivery of content to User Agents by Surrogates in one CDN (the downstream CDN) on behalf of another CDN (the upstream CDN).

Authoritative CDN: A CDN which has a direct relationship with a CSP for the distribution & delivery of that CSP’s content by the authoritative CDN or by downstream CDNs of the authoritative CDN.

Upstream CDN: For a given End User request, the CDN (within a pair of directly interconnected CDNs) that redirects the request to the other CDN.

Downstream CDN: For a given End User request, the CDN (within a pair of directly interconnected CDNs) to which the request is redirected by the other CDN (the Upstream CDN). Note that in the case of successive redirections (e.g. CDN1-->CDN2-->CDN3) a given CDN (e.g. CDN2) may act as the Downstream CDN for a redirection (e.g. CDN1-->CDN2) and as the Upstream CDN for the subsequent redirection of the same request (e.g. CDN2-->CDN3).

Over-the-top (OTT): A service, e.g. content delivery using a CDN, operated by a different operator than the NSP to which the users of that service are attached.

Surrogate: A device/function (often called a cache) that interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content. Typically, surrogates will cache requested content so that it can deliver the same content to a number of User Agents (and their End Users) avoiding the need for those requests to transit multiple times through the network core (i.e from the content origin to the surrogate).

Request Routing System: The function within a CDN responsible for receiving a content request from a User Agent, obtaining and

maintaining necessary information about a set of candidate surrogates or candidate CDNs, and for selecting and redirecting the user to the appropriate surrogate or CDN. To enable CDN Interconnection, the Request Routing System must also be capable of handling User Agent content requests passed to it by another CDN.

Distribution System: The function within a CDN responsible for distributing Content Distribution Metadata as well as the Content itself inside the CDN (e.g. down to the surrogates).

Delivery: The function within CDN surrogates responsible for delivering a piece of content to the User Agent. For example, delivery may be based on HTTP progressive download or HTTP adaptive streaming.

Logging System: The function within a CDN responsible for collecting the measurement and recording of distribution and delivery activities. The information recorded by the logging system may be used for various purposes including charging (e.g. of the CSP), analytics and monitoring.

Control System: The function within a CDN responsible for bootstrapping and controlling the other components of the CDN as well as for handling interactions with external systems (e.g. handling delivery service creation/update/removal requests, or specific service provisioning requests).

Quality of Experience (QoE): As defined in Section 2.4 of [RFC6390]

1.2. CDN Background

Readers are assumed to be familiar with the architecture, features and operation of CDNs. For readers less familiar with the operation of CDNs, the following resources may be useful:

- RFC 3040 [RFC3040] describes many of the component technologies that are used in the construction of a CDN.
- Taxonomy [TAXONOMY] compares the architecture of a number of CDNs.
- RFC 3466 [RFC3466] and RFC 3570 [RFC3570] are the output of the IETF Content Delivery Internetworking (CDI) working group which was closed in 2003.

Note: Some of the terms used in this document are similar to terms used the above referenced documents. When reading this document terms should be interpreted as having the definitions provided in Section 1.1.

2. CDN Interconnection Use Cases

An increasing number of NSPs are deploying CDNs in order to deal cost-effectively with the growing usage of on-demand video services and other content delivery applications.

CDNs allow caching of content closer to the edge of a network so that a given item of content can be delivered by a CDN Surrogate (i.e. a cache) to multiple User Agents (and their End Users) without transiting multiple times through the network core (i.e from the content origin to the surrogate). This contributes to bandwidth cost reductions for the NSP and to improved quality of experience for the End Users. CDNs also enable replication of popular content across many surrogates, which enables content to be served to large numbers of User Agents concurrently. This also helps dealing with situations such as flash crowds and denial of service attacks.

The CDNs deployed by NSPs are not just restricted to the delivery of content to support the Network Service Provider's own 'walled garden' services, such as IP delivery of television services to Set Top Boxes, but are also used for delivery of content to other devices including PCs, tablets, mobile phones etc.

Some service providers operate over multiple geographies and federate multiple affiliate NSPs. These NSPs typically operate independent CDNs. As they evolve their services (e.g. for seamless support of content services to nomadic users across affiliate NSPs) there is a need for interconnection of these CDNs, that represents a first use case for CDNI. However there are no open specifications, nor common best practices, defining how to achieve such CDN interconnection.

CSPs have a desire to be able to get (some of) their content to very large numbers of End Users, who are often distributed across a number of geographies, while maintaining a high quality of experience, all without having to maintain direct business relationships with many different CDN Providers (or having to extend their own CDN to a large number of locations). Some NSPs are considering interconnecting their respective CDNs (as well as possibly over-the-top CDNs) so that this collective infrastructure can address the requirements of CSPs in a cost effective manner. This represents a second use case for CDNI. In particular, this would enable the CSPs to benefit from on-net delivery (i.e. within the Network Service Provider's own network/ CDN footprint) whenever possible and off-net delivery otherwise, without requiring the CSPs to maintain direct business relationships with all the CDNs involved in the delivery. Again, CDN Providers (NSPs or over-the-top CDN operators) are faced with a lack of open specifications and best practices.

NSPs have often deployed CDNs as specialized cost-reduction projects within the context of a particular service or environment. Some NSPs operate separate CDNs for separate services. For example, there may be a CDN for managed IPTV service delivery, a CDN for web-TV delivery and a CDN for video delivery to Mobile terminals. As NSPs integrate their service portfolio, there is a need for interconnecting these CDNs, representing a third use case for CDNI. Again, NSPs face the problem of lack of open interfaces for CDN interconnection.

For operational reasons (e.g. disaster, flash crowd) or commercial reasons, an over-the-top CDN may elect to make use of another CDN (e.g. an NSP CDN with on-net Surrogates for a given footprint) for serving a subset of the user requests (e.g. requests from users attached to that NSP), which results in a fourth use case for CDNI because CDN Providers (over-the-top CDN Providers or NSPs) are faced with a lack of open specifications and best practices.

Use cases for CDN Interconnection are further discussed in [I-D.ietf-cdni-use-cases].

3. CDN Interconnection Model & Problem Area for IETF

This section discusses the problem area for the IETF work on CDN Interconnection.

Interconnecting CDNs involves interactions among multiple different functions and components that form each CDN. Only some of those require standardization.

Some NSPs have started to perform experiments to explore whether their CDN use cases can already be addressed with existing CDN implementations. One set of such experiments is documented in [I-D.bertrand-cdni-experiments]. The conclusions of those experiments are that while some basic limited CDN Interconnection functionality can be achieved with existing CDN technology, the current lack of any standardized CDNI interfaces with the necessary level of functionality such as those discussed in this document is preventing the deployment of CDN Interconnection.

Listed below are the four interfaces required to interconnect a pair of CDNs and that constitute the problem space of CDN Interconnection along with the required functionality of each interface for which standards do not currently exist. As part of the development of the CDNI interfaces it will also be necessary to agree on common mechanisms for how to identify and name the data objects that are to be interchanged between interconnected CDNs.

The use of the term "interface" is meant to encompass the protocol over which CDNI data representations (e.g. CDNI Metadata objects) are exchanged as well as the specification of the data representations themselves (i.e. what properties/fields each object contains, its structure, etc.).

- CDNI Control interface: This interface allows the "CDNI Control" system in interconnected CDNs to communicate. This interface may support the following:
  - Allow bootstrapping of the other CDNI interfaces (e.g. interface address/URL discovery and establishment of security associations).
  - Allow configuration of the other CDNI interfaces (e.g. Upstream CDN specifies information to be reported through the CDNI Logging interface).
  - Allow the downstream CDN to communicate static (or fairly static) information about its delivery capabilities and policies.
  - Allow bootstrapping of the interface between CDNs for content acquisition (even if that interface itself is outside the scope of the CDNI work).
  - Allow an upstream CDN to initiate or request specific actions to be undertaken in the downstream CDN. For example, to allow an upstream CDN to initiate content or CDNI Metadata acquisition (pre-positioning) or to request the invalidation or purging of content files and/or CDNI Metadata in a downstream CDN.
- CDNI Request Routing interface: This interface allows the Request Routing systems in interconnected CDNs to communicate to ensure that an End User request can be (re)directed from an upstream CDN to a surrogate in the downstream CDN, in particular where selection responsibilities may be split across CDNs (for example the upstream CDN may be responsible for selecting the downstream CDN while the downstream CDN may be responsible for selecting the actual surrogate within that downstream CDN). In particular, the functions of the CDN Request Routing interface may be divided as follows:
  - A CDNI Request Routing Redirection interface which allows the upstream CDN to query the downstream CDN at request routing time before redirecting the request to the downstream CDN.
  - A CDNI Footprint & Capabilities advertisement interface which allows the downstream CDN to provide to the upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the downstream CDN by the upstream CDN request routing system when processing subsequent content requests from User Agents.

- o CDNI Metadata distribution interface: This interface allows the Distribution system in interconnected CDNs to communicate to ensure CDNI Metadata can be exchanged across CDNs. See Section 1.1 for definition and examples of CDNI Metadata.
- o CDNI Logging interface: This interface allows the Logging system in interconnected CDNs to communicate the relevant activity logs in order to allow log consuming applications to operate in a multi-CDN environments. For example, an upstream CDN may collect delivery logs from a downstream CDN in order to perform consolidated charging of the CSP or for settlement purposes across CDNs. Similarly, an upstream CDN may collect delivery logs from a downstream CDN in order to provide consolidated reporting and monitoring to the CSP.

Note that the actual grouping of functionalities under these four interfaces is considered tentative at this stage and may be changed after further study (e.g. some subset of functionality be moved from one interface into another).

The above list covers a significant potential problem space, in part because in order to interconnect two CDNs there are several 'touch points' that require standardization. However, it is expected that the CDNI interfaces need not be defined from scratch and instead can very significantly reuse or leverage existing protocols; this is discussed further in Section 4.

The interfaces that form the CDNI problem area are illustrated in Figure 1.

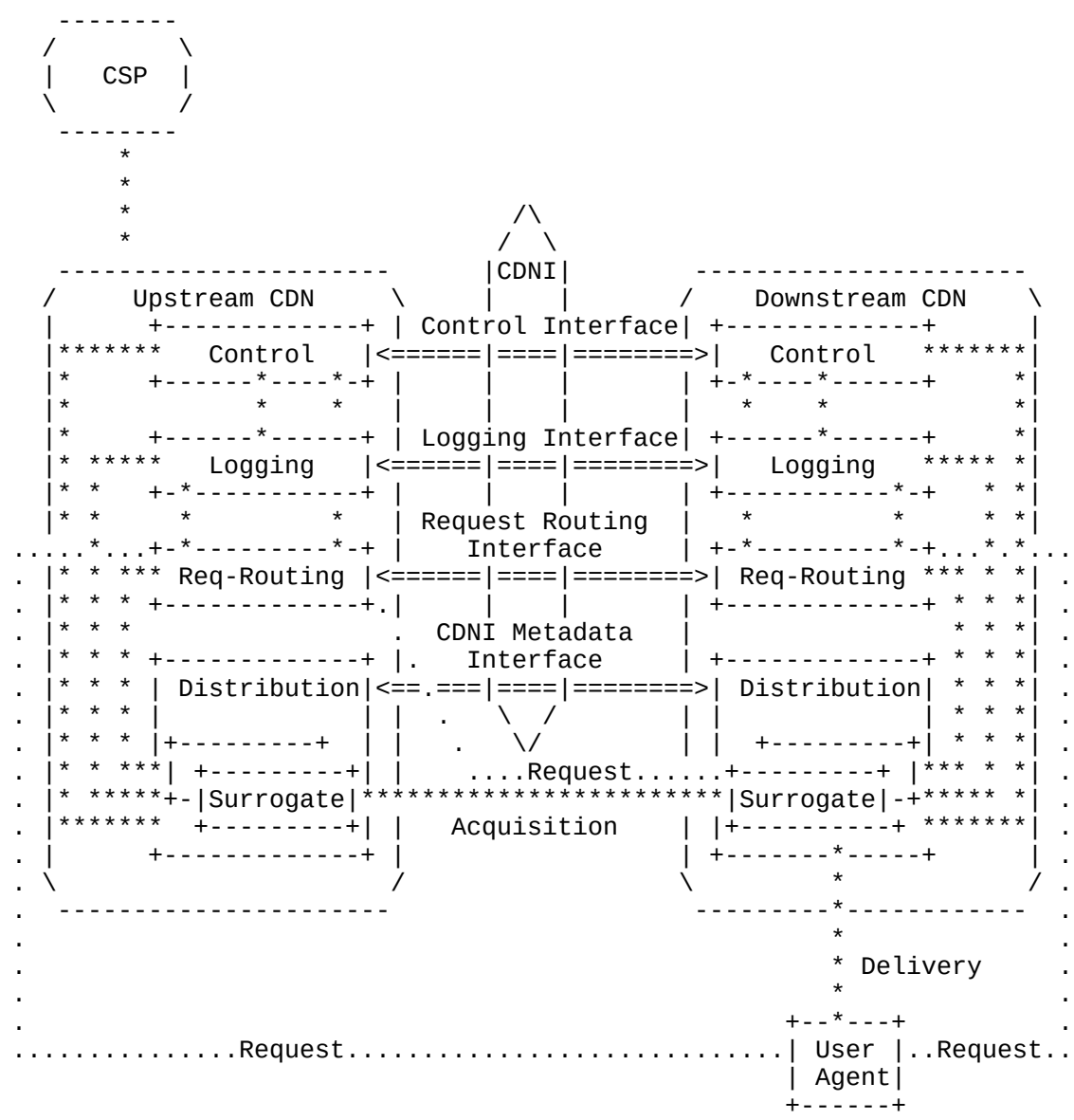


Figure 1: A Model for the CDNI Problem Area

As illustrated in Figure 1, the acquisition of content between

interconnected CDNs is out of scope for CDNI, which deserves some additional explanation. The consequence of such a decision is that the CDNI problem space described in this document is focussed on only defining the control plane for CDNI; and the CDNI data plane (i.e. the acquisition & distribution of the actual content objects) is out of scope. The rationale for such a decision is that CDNs today typically already use standardized protocols such as HTTP, FTP, rsync, etc. to acquire content from their CSP customers and it is expected that the same protocols could be used for acquisition between interconnected CDNs. Therefore the problem of content acquisition is considered already solved and all that is required from specifications developed by the CDNI working group is to describe within the CDNI Metadata the parameters to use to retrieve the content for example the IP address/hostname to connect to, what protocol to use to retrieve the content, etc.

4. Scoping the CDNI Problem

This section outlines how the scope of work addressing the CDNI problem space can be constrained through reuse or leveraging of existing protocols to implement the CDNI interfaces. This discussion is not intended to pre-empt any working group decision as to the most appropriate protocols, technologies and solutions to select to realize the CDNI interfaces but is intended as an illustration of the fact that the CDNI interfaces need not be created in a vacuum and that reuse or leverage of existing protocols is likely possible.

The four CDNI interfaces (CDNI Control interface, CDNI Request Routing interface, CDNI Metadata interface, CDNI Logging interface) described in Section 3 within the CDNI problem area are all control plane interfaces operating at the application layer (Layer 7 in the OSI network model). Firstly, since it is not expected that these interfaces would exhibit unique session, transport or network requirements as compared to the many other existing applications in the Internet, it is expected that the CDNI interfaces will be defined on top of existing session, transport and network protocols.

Secondly, although a new application protocol could be designed specifically for CDNI our analysis below shows that this is unnecessary and it is recommended that existing application protocols be reused or leveraged (HTTP [RFC2616], Atom Publishing Protocol [RFC5023], XMPP [RFC6120], for example) to realize the CDNI interfaces.

4.1. CDNI Request Routing Interface

The CDNI Request Routing interface enables a Request Routing function in an upstream CDN to query a Request Routing function in a downstream CDN to determine if the downstream CDN is able (and willing) to accept the delegated content request. It also allows the downstream CDN to control what should be returned to the User Agent in the redirection message by the upstream Request Routing function .

The CDNI Request Routing interface is therefore a fairly straightforward request/response interface and could be implemented over any number of request/response protocols. For example, it may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.). This removes the need for the CDNI working group to define a new protocol for the request/response element of the CDNI Request Routing interface.

Additionally, as discussed in Section 3, the CDNI Request Routing interface is also expected to enable a downstream CDN to provide to the upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the downstream CDN by the upstream CDN request routing system when processing subsequent content requests from User Agents. It is expected that such functionality of the CDNI request Routing could be specified by the CDNI working group with significant leveraging of existing IETF protocols supporting the dynamic distribution of reachability information (for example by leveraging existing routing protocols) or supporting application level queries for topological information (for example by leveraging ALTO [RFC5693]).

4.2. CDNI Metadata Interface

The CDNI Metadata interface enables the Distribution System in a downstream CDN to request CDNI Metadata from an upstream CDN so that the downstream CDN can properly process and respond to redirection requests received over the CDNI Request Routing interface and Content Requests received directly from User Agents.

The CDNI Metadata interface is therefore similar to the CDNI Request Routing interface because it is a request/response interface with the potential addition that CDNI Metadata search may have more complex semantics than a straightforward Request Routing redirection request. Therefore, like the CDNI Request Routing interface, the CDNI Metadata interface may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.) or possibly using other existing protocols such as XMPP [RFC6120]. This removes the need for the CDNI working group to define a new

protocol for the request/response element of the CDNI Metadata interface.

4.3. CDNI Logging Interface

The CDNI Logging interface enables details of content distribution and delivery activities to be exchanged between interconnected CDNs. For example the exchange of log records related to the delivery of content, similar to the log records recorded in a web server's access log.

Several protocols already exist that could potentially be used to exchange CDNI logs between interconnected CDNs including SNMP, syslog, ftp (and secure variants), HTTP POST, etc.

4.4. CDNI Control Interface

The CDNI Control interface allows the Control System in interconnected CDNs to communicate. The exact inter-CDN control functionality required to be supported by the CDNI Control interface is less well defined than the other three CDNI interfaces at this time.

It is expected that for the Control interface, as for the other CDNI Interfaces, existing protocols can be reused or leveraged.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

Distribution of content by a CDN comes with a range of security considerations such as how to enforce control of access to the content by end users in line with the CSP policy, or how to trust the logging information generated by the CDN for the purposes of charging the CSP. These security aspects are already dealt with by CDN Providers and CSPs today in the context of standalone CDNs. However, interconnection of CDNs introduces a new set of security considerations by extending the trust model to a chain of trust (i.e. the CSP "trusts" a CDN that "trusts" another CDN). The mechanisms used to mitigate these risks in multi-CDN environments may be similar to those used in the single CDN case, but their suitability in this

more complex environment must be validated.

The interconnection of CDNs may also introduce additional privacy considerations on top of those that apply to the single CDN case. In a multi-CDN environment, the different CDNs may reside in different legal regimes that require differing privacy requirements to be enforced. Such privacy requirements may impact the granularity of information that can be exchanged across the CDNI interfaces. For example the Logging System in a downstream CDN may need to apply some degree of anonymization, obfuscation or even the complete removal of some fields before exchanging log records containing details of End User deliveries with an upstream CDN.

Maintaining the security of the content itself, its associated metadata (including delivery policies) and the CDNs distributing and delivering it, are critical requirements for both CDN Providers and CSPs and the CDN Interconnection interfaces must provide sufficient mechanisms to maintain the security of the overall system of interconnected CDNs as well as the information (content, metadata, logs, etc) distributed and delivered through any set of interconnected CDNs.

6.1. Security of the CDNI Control interface

Information exchanged between interconnected CDNs over this interface is of a sensitive nature. A pair of CDNs use this interface to allow bootstrapping of all the other CDNI interfaces possibly including establishment of the mechanisms for securing these interfaces. Therefore, corruption of that interface may result in corruption of all other interfaces. Using this interface, an upstream CDN may pre-position or delete content or metadata in a downstream CDN and a downstream CDN may provide administrative information to an upstream CDN, etc. All of these operations require that the peer CDNs are appropriately authenticated and that the confidentiality and integrity of information flowing between them can be ensured.

6.2. Security of the CDNI Request Routing Interface

Appropriate levels of authentication and confidentiality must be used in this interface because it allows an upstream CDN to query the downstream CDN in order to redirect requests, and conversely, allows the downstream CDN to influence the upstream CDN’s Request Routing function.

In the absence of appropriate security on this interface, a rogue upstream CDN could inundate downstream CDNs with bogus requests, or have the downstream CDN send the rogue upstream CDN private information. Also, a rogue downstream CDN could influence the

upstream CDN so the upstream CDN redirects requests to the rogue dCDN or another dCDN in order to, for example, attract additional delivery revenue.

6.3. Security of the CDNI Metadata interface

This interface allows a downstream CDN to request CDNI metadata from an upstream CDN, and therefore the upstream CDN must ensure that the former is appropriately authenticated before sending the data. Conversely, a downstream CDN must authenticate an upstream CDN before requesting metadata to insulate itself from poisoning by rogue upstream CDNs. The confidentiality and integrity of the information exchanged between the peers must be protected.

6.4. Security of the CDNI Logging interface

Logging data consists of potentially sensitive information (which end user accessed which media resource, IP addresses of end users, potential names and subscriber account information, etc.). Confidentiality of this information must be protected as log records are moved between CDNs. This information may also be sensitive from the viewpoint that it can be the basis for charging across CDNs. Therefore, appropriate levels of protection are needed against corruption, duplication and loss of this information.

7. Acknowledgements

The authors would like to thank Andre Beck, Gilles Bertrand, Mark Carlson, Bruce Davie, David Ferguson, Yiu Lee, Kent Leung, Will Li, Kevin Ma, Julien Maisonneuve, Guy Meador, Larry Peterson, Emile Stephan, Oskar van Deventer, Mahesh Viveganandhan and Richard Woundy for their review comments and contributions to the text.

8. References

8.1. Normative References

8.2. Informative References

[3GP-DASH] "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) http://www.3gpp.org/ftp/Specs/html-info/26247.htm".

[ALTO-Charter]

"IETF ALTO WG Charter
(http://datatracker.ietf.org/wg/alto/charter/)".

[ATIS] "ATIS (http://www.atis.org/)".

[ATIS-COD]
"ATIS IIF: IPTV Content on Demand Service, January 2011 (h
ttp://www.atis.org/iif/_Com/Docs/Task_Forces/ARCH/
ATIS-0800042.pdf)".

[CDI-Charter]
"IETF CDI WG Charter
(http://www.ietf.org/wg/concluded/cdi)".

[CableLabs]
"CableLabs (http://www.cablelabs.com/about/)".

[CableLabs-Metadata]
"CableLabs VoD Metadata Project Primer
(http://www.cablelabs.com/projects/metadata/primer/)".

[DECADE-Charter]
"IETF DECADE WG Charter
(http://datatracker.ietf.org/wg/decade/charter/)".

[I-D.bertrand-cdni-experiments]
Faucheur, F. and L. Peterson, "Content Distribution
Network Interconnection (CDNI) Experiments",
draft-bertrand-cdni-experiments-02 (work in progress),
February 2012.

[I-D.ietf-cdni-use-cases]
Bertrand, G., Emile, S., Burbridge, T., Eardley, P., Ma,
K., and G. Watson, "Use Cases for Content Delivery Network
Interconnection", draft-ietf-cdni-use-cases-08 (work in
progress), June 2012.

[I-D.jenkins-alto-cdn-use-cases]
Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and
S. Previdi, "Use Cases for ALTO within CDNs",
draft-jenkins-alto-cdn-use-cases-03 (work in progress),
June 2012.

[MPEG-DASH]
"Information technology - MPEG systems technologies - Part
6: Dynamic adaptive streaming over HTTP (DASH), (DIS
version), February 2011
http://mpeg.chiariglione.org/

working_documents.htm#MPEG-B".

[OIPF-Overview]
"OIPF Release 2 Specification Volume 1 - Overview", September 2010.

[P2PRG-CDNI]
Davie, B. and F. Le Faucheur, "Interconnecting CDNs aka "Peering Peer-to-Peer" (http://www.ietf.org/proceedings/77/slides/P2PRG-2.pdf)", March 2010.

[PPSP-Charter]
"IETF PPSP WG Charter (http://datatracker.ietf.org/wg/ppsp/charter/)".

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3040] Cooper, I., Melve, I., and G. Tomlinson, "Internet Web Replication and Caching Taxonomy", RFC 3040, January 2001.

[RFC3466] Day, M., Cain, B., Tomlinson, G., and P. Rzewski, "A Model for Content Internetworking (CDI)", RFC 3466, February 2003.

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

[RFC3570] Rzewski, P., Day, M., and D. Gilletti, "Content Internetworking (CDI) Scenarios", RFC 3570, July 2003.

[RFC5023] Gregorio, J. and B. de hOra, "The Atom Publishing Protocol", RFC 5023, October 2007.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.

[SNIA-CDMI]
"SNIA CDMI (http://www.snia.org/tech_activities/standards/curr_standards/cdmi)".

[TAXONOMY]
Pathan, A., "A Taxonomy and Survey of Content Delivery Networks (http://www.gridbus.org/reports/CDN-Taxonomy.pdf)", 2007.

[Y.1910] "ITU-T Recomendation Y.1910 "IPTV functional architecture"", September 2008.

[Y.2019] "ITU-T Recomendation Y.2019 "Content delivery functional architecture in NGN"", September 2010.

Appendix A. Design considerations for realizing the CDNI Interfaces

This section expands on how CDNI interfaces can reuse and leverage existing protocols before describing each CDNI interface individually and highlighting example candidate protocols that could be considered for reuse or leveraging to implement the CDNI interfaces.

A.1. CDNI Request Routing Interface

The CDNI Request Routing interface enables a Request Routing function in an upstream CDN to query a Request Routing function in a downstream CDN to determine if the downstream CDN is able (and willing) to accept the delegated content request and to allow the downstream CDN to control what the upstream Request Routing function should return to the User Agent in the redirection message.

Therefore, the CDNI Request Routing interface needs to offer a mechanism for an upstream CDN to issue a "Redirection Request" to a downstream CDN. The Request Routing interface needs to be able to support scenarios where the initial User Agent request to the upstream CDN is received over DNS as well as over a content specific application protocol (e.g. HTTP, RTSP, RTMP, etc.).

Therefore a Redirection Request is expected to contain information such as:

- The protocol (e.g. DNS, HTTP) over which the upstream CDN received the initial User Agent request.
- Additional details of the User Agent request that are required to perform effective Request Routing by the Downstream CDN. For DNS this would typically be the IP address of the DNS resolver making the request on behalf of the User Agent. For requests received

over content specific application protocols the Redirection Request could contain significantly more information related to the original User Agent request but at a minimum is expected to include the User Agent's IP address, the equivalent of the HTTP Host header and the equivalent of the HTTP abs_path defined in [RFC2616].

It should be noted that, the CDNI architecture needs to consider that a downstream CDN may receive requests from User Agents without first receiving a Redirection Request from an upstream CDN for the corresponding User Agent request, for example because:

- o User Agents (or DNS resolvers) may cache DNS or application responses from Request Routers.
- o Responses to Redirection Requests over the Request Routing interface may be cacheable.
- o Some CDNs may rely on simple coarse policies, e.g. CDN B agrees to always serve CDN A's delegated redirection requests, in which case the necessary redirection details are exchanged out of band (of the CDNI interfaces), e.g. configured.

On receiving a Redirection Request, the downstream CDN will use the information provided in the request to determine if it is able (and willing) to accept the delegated content request and needs to return the result of its decision to the upstream CDN.

Thus, a Redirection Response from the downstream CDN is expected to contain information such as:

- o Status code indicating acceptance or rejection (possibly with accompanying reasons).
- o Information to allow redirection by the Upstream CDN. In the case of DNS-based request routing, this is expected to include the equivalent of a DNS record(s) (e.g. a CNAME) that the upstream CDN should return to the requesting DNS resolver. In the case of application based request routing, this is expected to include the information necessary to construct the application specific redirection response(s) to return to the requesting User Agent. For HTTP requests from User Agents this could include a URI that the upstream CDN could return in a HTTP 3xx response.

The CDNI Request Routing interface is therefore a fairly straightforward request/response interface and could be implemented over any number of request/response protocols. For example, it may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.). This removes the need for the CDNI working group to define a new protocol for the request/response element of the CDNI Request Routing

interface. Thus, the CDNI working group would be left only with the task of specifying:

- The recommended request/response protocol to use along with any additional semantics and procedures that are specific to the CDNI Request Routing interface (e.g. handling of malformed requests/ responses).
- The syntax (i.e representation/encoding) of the redirection requests and responses.
- The semantics (i.e. meaning and expected contents) of the redirection requests and responses.

Additionally, as discussed in Section 3, the CDNI Request Routing interface is also expected to enable a downstream CDN to provide to the upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the downstream CDN by the upstream CDN request routing system when processing subsequent content requests from User Agents. It is expected that such functionality of the CDNI request Routing could be specified by the CDNI working group with significant leveraging of existing IETF protocols supporting the dynamic distribution of reachability information (for example by leveraging existing routing protocols) or supporting application level queries for topological information (for example by leveraging ALTO).

A.2. CDNI Metadata Interface

The CDNI Metadata interface enables the Distribution System in a downstream CDN to obtain CDNI Metadata from an upstream CDN so that the downstream CDN can properly process and respond to:

- Redirection Requests received over the CDNI Request Routing interface.
- Content Requests received directly from User Agents.

The CDNI Metadata interface needs to offer a mechanism for an Upstream CDN to:

- Distribute/update/remove CDNI Metadata to a Downstream CDN.

and/or to allow a downstream CDN to:

- Make direct requests for CDNI Metadata objects
- Make recursive requests for CDNI metadata, for example to enable a downstream CDN to walk down a tree of objects with inter-object relationships.

The CDNI Metadata interface is therefore similar to the CDNI Request

Routing interface because it is a request/response interface with the potential addition that CDNI Metadata search may have more complex semantics than a straightforward Request Routing redirection request. Therefore, like the CDNI Request Routing interface, the CDNI Metadata interface may be implemented as a WebService using one of the common WebServices methodologies (XML-RPC, HTTP query to a known URI, etc.) or possibly using other existing protocols such as XMPP [RFC6120]. This removes the need for the CDNI working group to define a new protocol for the request/response element of the CDNI Metadata interface.

Thus, the CDNI working group would be left only with the task of specifying:

- The recommended request/response protocol to use along with any additional semantics that are specific to the CDNI Metadata interface (e.g. handling of malformed requests/responses).
- The syntax (i.e representation/encoding) of the CDNI Metadata objects that will be exchanged over the interface.
- The semantics (i.e. meaning and expected contents) of the individual properties of a Metadata object.
- How the relationships between different CDNI Metadata objects are represented.

A.3. CDNI Logging Interface

The CDNI Logging interface enables details of content distribution and delivery activities to be exchanged between interconnected CDNs, such as log records related to the delivery of content (similar to the log records recorded in a web server’s access log).

Within CDNs today, log records are used for a variety of purposes. Specifically CDNs use logs to generate Call Data Records (CDRs) for passing to billing and payment systems and to real-time (and near real-time) analytics systems. Such applications place requirements on the CDNI Logging interface to support guaranteed and timely delivery of log messages between interconnected CDNs. It may also be necessary to be able to prove the integrity of received log messages.

Several protocols already exist that could potentially be used to exchange CDNI logs between interconnected CDNs including SNMP Traps, syslog, ftp, HTTP POST, etc. although it is likely that some of the candidate protocols may not be well suited to meet all the requirements of CDNI. For example SNMP traps pose scalability concerns and SNMP does not support guaranteed delivery of Traps and therefore could result in log records being lost and the consequent CDRs and billing records for that content delivery not being produced as well as that content delivery being invisible to any analytics

platforms.

Although it is not necessary to define a new protocol for exchanging logs across the CDNI Logging interface, the CDNI working group would still need to specify:

- The recommended protocol to use.
- A default set of log fields and their syntax & semantics. Today there is no standard set of common log fields across different content delivery protocols and in some cases there is not even a standard set of log field names and values for different implementations of the same delivery protocol.
- A default set of conditions that trigger log records to be generated.

A.4. CDNI Control Interface

The CDNI Control interface allows the Control System in interconnected CDNs to communicate. The exact inter-CDN control functionality required to be supported by the CDNI Control interface is less well defined than the other three CDNI interfaces at this time.

However, as discussed in Section 3, the CDNI Control interface may be required to support functionality similar to the following:

- Allow an upstream CDN and downstream CDN to establish, update or terminate their CDNI interconnection.
- Allow bootstrapping of the other CDNI interfaces (e.g. protocol address discovery and establishment of security associations).
- Allow configuration of the other CDNI interfaces (e.g. Upstream CDN specifies information to be reported through the CDNI Logging interface).
- Allow the downstream CDN to communicate static information about its delivery capabilities, resources and policies.
- Allow bootstrapping of the interface between CDNs for content acquisition (even if that interface itself is outside the scope of the CDNI work).

It is expected that for the Control interface also, existing protocols can be reused or leveraged. Those will be considered once the requirements for the Control interface have been refined.

Appendix B. Additional Material

This section records related information that was produced as part of defining the CDNI problem statement.

B.1. Non-Goals for IETF

Listed below are aspects of content delivery that the authors propose be kept outside of the scope of the CDNI working group:

- The interface between Content Service Provider and the Authoritative CDN (i.e. the upstream CDN contracted by the CSP for delivery by this CDN or by its downstream CDNs).
- The delivery interface between the delivering CDN surrogate and the User Agent, such as streaming protocols.
- The request interface between the User Agent and the request-routing system of a given CDN. Existing IETF protocols (e.g. HTTP, RTSP, DNS) are commonly used by User Agents to request content from a CDN and by CDN request routing systems to redirect the User Agent requests. The CDNI working group need not define new protocols for this purpose. Note however, that the CDNI control plane interface may indirectly affect some of the information exchanged through the request interface (e.g. URI).
- The content acquisition interface between CDNs (i.e. the data plane interface for actual delivery of a piece of content from one CDN to the other). This is expected to use existing protocols such as HTTP or protocols defined in other forums for content acquisition between an origin server and a CDN (e.g. HTTP-based C2 reference point of ATIS IIF CoD). The CDN Interconnection problem space described in this document may therefore only concern itself with the agreement/negotiation aspects of which content acquisition protocol is to be used between two interconnected CDNs in view of facilitating interoperability.
- End User/User Agent Authentication. End User/User Agent authentication and authorization are the responsibility of the Content Service Provider.
- Content preparation, including encoding and transcoding. The CDNI architecture aims at allowing distribution across interconnected CDNs of content treated as opaque objects. Interpretation and processing of the objects, as well as optimized delivery of these objects by the surrogate to the End User are outside the scope of CDNI.
- Digital Rights Management (DRM). DRM is an end-to-end issue between a content protection system and the User Agent.
- Applications consuming CDNI logs (e.g. charging, analytics, reporting,...).
- Internal CDN interfaces & protocols (i.e. interfaces & protocols within one CDN).
- Scalability of individual CDNs. While scalability of the CDNI interfaces/approach is in scope, how an individual CDN scales is out of scope.
- Actual algorithms for selection of CDNs or Surrogates by Request Routing systems (however, some specific parameters required as input to these algorithms may be in scope when they need to be

communicated across CDNs).
- o Surrogate algorithms. For example caching algorithms and content acquisition methods are outside the scope of the CDNI work. Content management (e.g. Content Deletion) as it relates to CDNI content management policies, is in scope but the internal algorithms used by a cache to determine when to no longer cache an item of Content (in the absence of any specific metadata to the contrary) is out of scope.
- o Element management interfaces.
- o Commercial, business and legal aspects related to the interconnections of CDNs.

B.2. Relationship to relevant IETF Working Groups & IRTF Reserach Groups

B.2.1. ALTO WG

As stated in the ALTO Working Group charter [ALTO-Charter]:

"The Working Group will design and specify an Application-Layer Traffic Optimization (ALTO) service that will provide applications with information to perform better-than-random initial peer selection. ALTO services may take different approaches at balancing factors such as maximum bandwidth, minimum cross-domain traffic, lowest cost to the user, etc. The working group will consider the needs of BitTorrent, tracker-less P2P, and other applications, such as content delivery networks (CDN) and mirror selection."

In particular, the ALTO service can be used by a CDN Request Routing system to improve its selection of a CDN surrogate to serve a particular User Agent request (or to serve a request from another surrogate). [I-D.jenkins-alto-cdn-use-cases] describes a number of use cases for a CDN to be able to obtain network topology and cost information from an ALTO server(s) and discusses how CDN Request Routing could be used as an integration point of ALTO into CDNs. It is possible that the ALTO service could be used in the same manner in a multi-CDN environment based on CDN Interconnection. For example, an upstream CDN may take advantage of the ALTO service in its decision for selecting a downstream CDN to which a user request should be delegated.

However, the current work of ALTO is complementary to and does not overlap with the work described in this document because the integration between ALTO and a CDN is an internal decision for a specific CDN and is therefore out of scope for the CDNI working group. One area for further study is whether additional information should be provided by an ALTO service to facilitate CDNI CDN selection.

B.2.2. DECADE WG

The DECADE Working Group [DECADE-Charter] is addressing the problem of reducing traffic on the last-mile uplink, as well as backbone and transit links caused by P2P streaming and file sharing applications. It addresses the problem by enabling an application endpoint to make content available from an in-network storage service and by enabling other application endpoints to retrieve the content from there.

Exchanging data through the in-network storage service in this manner, instead of through direct communication, provides significant gain where:

- The network capacity/bandwidth from in-network storage service to application endpoint significantly exceeds the capacity/bandwidth from application endpoint to application endpoint (e.g. because of an end-user uplink bottleneck); and
- Where the content is to be accessed by multiple instances of application endpoints (e.g. as is typically the case for P2P applications).

While, as is the case for any other data distribution application, the DECADE architecture and mechanisms could potentially be used for exchange of CDNI control plane information via an in-network-storage service (as opposed to directly between the entities terminating the CDNI interfaces in the neighbor CDNs), we observe that:

- CDNI would operate as a "Content Distribution Application" from the DECADE viewpoint (i.e. would operate on top of DECADE).
- There does not seem to be obvious benefits in integrating the DECADE control plane responsible for signaling information relating to control of the in-network storage service itself, and the CDNI control plane responsible for application-specific CDNI interactions (such as exchange of CDNI metadata, CDNI request redirection, transfer of CDNI logging information).
- There would typically be limited benefits in making use of a DECADE in-network storage service because the CDNI interfaces are expected to be terminated by a very small number of CDNI clients (if not one) in each CDN, and the CDNI clients are expected to benefit from high bandwidth/capacity when communicating directly to each other (at least as high as if they were communicating via an in-network storage server).

The DECADE in-network storage architecture and mechanisms may theoretically be used for the acquisition of the content objects themselves between interconnected CDNs. It is not expected that this would have obvious benefits in typical situations where a content object is acquired only once from an Upstream CDN to a Downstream CDN

(and then distributed as needed inside the Downstream CDN). But it might have benefits in some particular situations. Since the acquisition protocol between CDNs is outside the scope of the CDNI work, this question is left for further study.

The DECADE in-network storage architecture and mechanisms may potentially also be used within a given CDN for the distribution of the content objects themselves among surrogates of that CDN. Since the CDNI work does not concern itself with operation within a CDN, this question is left for further study.

Therefore, the work of DECADE may be complementary to but does not overlap with the CDNI work described in this document.

B.2.3. PPSP WG

As stated in the PPSP Working Group charter [PPSP-Charter]:

"The Peer-to-Peer Streaming Protocol (PPSP) working group develops two signaling and control protocols for a peer-to-peer (P2P) streaming system for transmitting live and time-shifted media content with near real-time delivery requirements." and "The PPSP working group designs a protocol for signaling and control between trackers and peers (the PPSP "tracker protocol") and a signaling and control protocol for communication among the peers (the PPSP "peer protocol"). The two protocols enable peers to receive streaming data within the time constraints required by specific content items."

Therefore PPSP is concerned with the distribution of the streamed content itself along with the necessary signaling and control required to distribute the content. As such, it could potentially be used for the acquisition of streamed content across interconnected CDNs. But since the acquisition protocol is outside the scope of the work proposed for CDNI, we leave this for further study. Also, because of its streaming nature, PPSP is not seen as applicable to the distribution and control of the CDNI control plane and CDNI data representations.

Therefore, the work of PPSP may be complementary to but does not overlap with the work described in this document for CDNI.

B.2.4. IRTF P2P Research Group

Some information on CDN interconnection motivations and technical issues were presented in the P2P RG at IETF 77. The presentation can be found in [P2PRG-CDNI].

Appendix C. Additional Material

Note to RFC Editor: This appendix is to be removed on publication as an RFC.

C.1. Related standardization activites

There are a number of other standards bodies and industry forums that are working in areas related to CDNs, and in some cases related to CDNI. This section outlines any potential overlap with the work of the CDNI working group and any component that could potentially be reused to realize the CDNI interfaces.

A number of standards bodies have produced specifications related to CDNs, for example:

- ETSI TISPAN (Telecommunications and Internet converged Services and Protocols for Advanced Networking) has a series of specifications focusing on CDNs.
- The Open IPTV Forum (OIPF) and ATIS IPTV Interoperability Forum (IIF) specify the architecture and the protocols of an IPTV solution. Although OIPF and ATIS specifications include the interaction with a CDN, the CDN specifications are coupled with their IPTV specifications and do not cover interconnection of CDNs.
- ATIS Cloud Services Forum (CSF) has started investigating interconnection of CDNs. The ATIS CSF focuses on defining use cases and requirements for such CDN interconnection, which are expected to be considered as input into the work of the CDNI working group. At the time of writing this document, ATIS CSF is not specifying the corresponding protocols or interfaces and is expected to leverage the work of the IETF CDNI working group for those.
- CableLabs, SNIA and ITU have developed (or are working on) definitions for content related metadata and specifications for its distribution. However, they do not include metadata specific to the distribution of content within a CDN or between interconnected CDNs.
- IETF CDI working group (now concluded) touched on the same problem space as the present document. However, in accordance with its initial charter, the CDI working group did not define any protocols or interfaces to actually enable CDN Interconnection and at that time (2003) there was not enough industry interest and real life requirements to justify rechartering the working group to conduct the corresponding protocol work.

Although some of the specifications describe multi-CDN cooperation or include reference points for interconnecting CDNs, none of them

specify in sufficient detail all the CDNI interfaces and CDNI Metadata representations required to enable even a base level of CDN Interconnection functionality to be implemented.

C.1.1. IETF CDI Working Group (Concluded)

The Content Distribution Internetworking (CDI) Working Group was formed in the IETF following a BoF in December 2000 and closed in mid 2003.

For convenience, here is an extract from the CDI working group charter [CDI-Charter]:

"

- The goal of this working group is to define protocols to allow the interoperation of separately-administered content networks.
- A content network is an architecture of network elements, arranged for efficient delivery of digital content. Such content includes, but is not limited to, web pages and images delivered via HTTP, and streaming or continuous media which are controlled by RTSP.
- The working group will first define requirements for three modes of content internetworking: interoperation of request-routing systems, interoperation of distribution systems, and interoperation of accounting systems. These requirements are intended to lead to a follow-on effort to define protocols for interoperation of these systems.
- In its initial form, the working group is not chartered to deliver those protocols [...]

"

Thus, the CDI working group touched on the same problem space as the present document.

The CDI working group published 3 Informational RFCs:

- RFC 3466 [RFC3466] - "A Model for Content Internetworking (CDI)".
- RFC 3568 [RFC3568] - "Known Content Network (CN) Request-Routing Mechanisms".
- RFC 3570 [RFC3570] - "Content Internetworking (CDI) Scenarios".

C.1.2. 3GPP

3GPP was the first organization that released a specification related to adaptive streaming over HTTP. 3GPP Release 9 specification on adaptive HTTP streaming was published in March 2010, and there have been some bug fixes on this specification since the publication. In

addition, 3GPP has produced an extended version for Release 10, which was published in 2011. This release will include a number of clarifications, improvements and new features.

[3GP-DASH] is defined as a general framework independent of the data encapsulation format. It has support for fast initial startup and seeking, adaptive bitrate switching, re-use of HTTP origin and cache servers, re-use of existing media playout engines, on-demand, live and time-shifted delivery. It specifies syntax and semantics of Media Presentation Description (MPD), format of segments and delivery protocol for segments. It does not specify content provisioning, client behavior or transport of MPD.

The content retrieved by a client using [3GP-DASH] adaptive streaming could be obtained from a CDN but this is not discussed or specified in the 3GPP specifications as it is transparent to [3GP-DASH] operations. Similarly, it is expected that [3GP-DASH] can be used transparently from the CDNs as a delivery protocol (between the delivering CDN surrogate and the User Agent) in a CDN Interconnection environment. [3GP-DASH] could also be a candidate for content acquisition between CDNs in a CDN Interconnection environment.

C.1.3. ISO MPEG

Within ISO MPEG, the Dynamic Adaptive Streaming over HTTP (DASH) ad-hoc group adopted the 3GPP Release 9 [3GP-DASH] specification as a starting point and has made some improvements and extensions. Similar to 3GPP SA4, the MPEG DASH ad-hoc group has been working on standardizing the manifest file and the delivery format. Additionally, the MPEG DASH ad-hoc group has also been working on the use of MPEG-2 Transport Streams as a media format, conversion from/to existing file formats, common encryption, and so on. The MPEG DASH specification could also be a candidate for delivery to the User Agent and for content acquisition between CDNs in a CDN Interconnection environment. The Draft International Standard (DIS) version [MPEG-DASH] is currently publicly available since early February 2011.

In the 95th MPEG meeting in January 2011, the DASH ad-hoc group decided to start a new evaluation experiment called "CDN-EE". The goals are to understand the requirements for MPEG DASH to better support CDN-based delivery, and to provide a guidelines document for CDN operators to better support MPEG DASH streaming services. The ongoing work is still very preliminary and does not currently target looking into CDN Interconnection use cases.

C.1.4. ATIS IIF

ATIS ([ATIS]) IIF is the IPTV Interoperability Forum (within ATIS) that develops requirements, standards, and specifications for IPTV.

ATIS IIF is developing the "IPTV Content on Demand (CoD) Service" specification. This includes use of a CDN (referred to in ATIS IIF CoD as the "Content Distribution and Delivery Functions") for support of a Content on Demand (CoD) Service as part of a broader IPTV service. However, this only covers the case of a managed IPTV service (in particular where the CDN is administered by the service provider) and does not cover the use, or interconnection, of multiple CDNs.

C.1.5. CableLabs

"Founded in 1988 by cable operating companies, Cable Television Laboratories, Inc. (CableLabs) is a non-profit research and development consortium that is dedicated to pursuing new cable telecommunications technologies and to helping its cable operator members integrate those technical advancements into their business objectives." [CableLabs]

CableLabs has defined specifications for CoD Content Metadata as part of its VOD Metadata project.

C.1.6. ETSI MCD

ETSI MCD (Media Content Distribution) is the ETSI technical committee "in charge of guiding and coordinating standardization work aiming at the successful overall development of multimedia systems (television and communication) responding to the present and future market requests on media content distribution".

MCD created a specific work item on interconnection of heterogeneous CDNs ("CDN Interconnection, use cases and requirements") in March 2010. MCD very recently created a working group to progress this work item. However, no protocol level work has yet started in MCD for CDN Interconnection.

C.1.7. ETSI TISPAN

ETSI TISPAN has published two sets of IPTV specifications, one of which is based on IMS. In addition, TISPAN has published a CDN architecture supporting delivery of various content services such as time-shifted TV and VoD to TISPAN devices (UEs) or regular PCs. The use cases allow for hierarchically and geographically distributed CDN scenarios, along with multi-CDN cooperation. As a result, the

architecture contains reference points to support interconnection of other TISPAN CDNs. The protocol definition phase for the corresponding CDN architecture was kicked-off at the end of 2010 as is still in progress. In line with its long history of leveraging IETF protocols, ETSI could potentially leverage CDNI interfaces developed in the IETF for their related protocol level work on interconnections of CDNs.

C.1.8. ITU-T

SG13 is developing standards related to the support of IPTV services (i.e.. multimedia services such as television/VoD/audio/text/ graphics/data delivered over IP-based managed networks).

ITU-T Recommendation Y.1910 [Y.1910] provides the description of the IPTV functional architecture. This architecture includes functions and interfaces for the distribution and delivery of content. This architecture is aligned with the ATIS IIF architecture.

Based upon ITU-T Rec. Y.1910, ITU-T Rec. Y.2019 [Y.2019] describes in more detail the content delivery functional architecture. This architecture allows CDN Interconnection: some interfaces (such as D3, D4) at the control level allow relationships between different CDNs, in the same domain or in different domains. Generic procedures are described, but the choice of the protocols is open.

C.1.9. Open IPTV Forum (OIPF)

The Open IPTV Forum has developed an end-to-end solution to allow any OIPF terminal to access enriched and personalized IPTV services either in a managed or a non-managed network[OIPF-Overview]. Some OIPF services (such as Network PVR) may be hosted in a CDN.

To that end, the Open IPTV Forum specification is made of 5 parts:

- Media Formats including HTTP Adaptive Streaming
- Content Metadata
- Protocols
- Terminal (Declarative or Procedural Application Environment)
- Authentication, Content Protection and Service Protection

C.1.10. TV-Anytime Forum

Version 1 of the TV-Anytime Forum specifications were published as ETSI TS 102 822-1 through ETSI TS 102 822-7 "Broadcast and On-line Services: Search, select, and rightful use of content on personal storage systems ("TV-Anytime")". It includes the specification of content metadata in XML schemas (ETSI TS 102 822-3) which define

technical parameters for the description of CoD and Live contents. The specification is referenced by DVB and OIPF.

The TV-anytime Forum was closed in 2005.

C.1.11. SNIA

The Storage Networking Industry Association (SNIA) is an association of producers and consumers of storage networking products whose goal is to further storage networking technology and applications.

SNIA has published the Cloud Data Management Interface (CDMI) standard ([SNIA-CDMI]).

"The Cloud Data Management Interface defines the functional interface that applications will use to create, retrieve, update and delete data elements from the Cloud. As part of this interface the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface."

C.1.12. Summary of existing standardization work

The following sections will summarize the existing work of the standard bodies listed earlier against the CDNI problem space. Appendix C.1.12.1 summarizes existing interfaces that could be leveraged for content acquisition between CDNs and Appendix C.1.12.2 summarizes existing metadata specifications that may be applicable to CDNI. To date we are not aware of any standardization activities in the areas of the remaining CDNI interfaces (CDNI Request Routing, CDNI Control and CDNI Logging).

C.1.12.1. Content Acquisition across CDNs and Delivery to End User (Data plane)

A number of standards bodies have completed work in the areas of content acquisition interface between a CSP and a CDN, as well as as on the delivery interface between the surrogate and the User Agent. Some of this work is summarized below.

TISPAN, OIPF and ATIS have specified IPTV and/or Content on Demand (CoD) services, including the data plane aspects (typically different flavors of RTP/RTCP and HTTP) to obtain content and deliver it to User Agents. For example, :

- The OIPF data plane includes both RTP and HTTP flavors (HTTP progressive download, HTTP Adaptive streaming [3GP-DASH]).

- The ATIS IIF specification "IPTV Content on Demand (CoD) Service" [ATIS-COD] defines a reference point (C2) and the corresponding HTTP-based data plane protocol for content acquisition between an authoritative origin server and the CDN.

While these protocols have not been explicitly specified for content acquisition across CDNs, they are suitable (in addition to others such as standard HTTP) for content acquisition between CDNs in a CDN Interconnection environment. Therefore for the purpose of the CDNI working group there are already multiple existing data plane protocols that can be used for content acquisition across CDNs.

Similarly, there are multiple existing standards (e.g. the OIPF data plane mentioned above, HTTP adaptive streaming [3GP-DASH]) or public specifications (e.g. vendor specific HTTP Adaptive streaming specifications) so that content delivery can be considered already solved (or at least sufficiently addressed in other forums).

Thus, specification of the content acquisition interface between CDNs and the delivery interface between the surrogate and the User Agent are out of scope for the CDNI working group. The CDNI working group may only concern itself with the negotiation/selection aspects of the acquisition protocol to be used in a CDN interonnect scenario.

C.1.12.2. CDNI Metadata

CableLabs, ITU, OIPF and TV-Anytime have work items dedicated to the specification of content metadata:

- CableLabs has defined specifications for CoD Content Metadata as part of its VOD Metadata project. "The VOD Metadata project is a cable television industry and cross-industry-wide effort to specify the metadata and interfaces for distribution of video-on-demand (VOD) material from multiple content providers to cable operators." [CableLabs-Metadata]. However, while the CableLabs work specifies an interface between a content provider and a service provider running a CDN, it does not include an interface that could be used between CDNs.
- ITU Study Group 16 has started work on a number of draft Recommendations (H.IPTV-CPMD, H.IPTV-CPMD, HSTP.IPTV-CMA, HSTP.IPTV-UMCI) specifying metadata for content distribution in IPTV services.
- An Open IPTV Terminal receives the technical description of the content distribution from the OIPF IPTV platform before receiving any content. The Content distribution metadata is sent in the format of a TV-Anytime XSD including tags to describes the location and program type (on demand or Live) as well as describing the time availability of the on demand and live content.

However the specifications outlined above do not include metadata specific to the distribution of content within a CDN or between interconnected CDNs, for example geo-blocking information, availability windows, access control mechanisms to be enforced by the surrogate, how to map an incoming content request to a file on the origin server or acquire it from the upstream CDN etc.

The CDMI standard ([SNIA-CDMI]) from SNIA defines metadata that can be associated with data that is stored by a cloud storage provider. While the metadata currently defined do not match the needs of CDN Interconnection, it is worth considering CDMI as one of the existing pieces of work that may potentially be leveraged for the CDNI Metadata interface (e.g by extending the CDMI metadata to address more specific CDNI needs).

C.2. Related Research Projects

C.2.1. OCEAN

OCEAN (http://www.ict-ocean.eu/) is an EU funded research project that started in February 2010 for 3 years. Some of its objectives are relevant to CDNI. It aims, among other things, at designing a new architectural framework for audiovisual content delivery over the Internet, defining public interfaces between its major building blocks in order to foster multi-vendor solutions and interconnection between Content Networks (the term "Content Networks" corresponds here to the definition introduced in [RFC3466], which encompasses CDNs).

OCEAN has not yet published any open specifications, nor common best practices, defining how to achieve such CDN interconnection.

C.2.2. Eurescom P1955

Eurescom P1955 was a 2010 research project involving a four European Network operators, which studied the interests and feasibility of interconnecting CDNs by firstly elaborating the main service models around CDN interconnection, as well as analyzing an adequate CDN interconnection technical architecture and framework, and finally by providing recommendations for telcos to implement CDN interconnection. The Eurescom P1955 project ended in July 2010.

The authors are not aware of material discussing CDN interconnection protocols or interfaces made publicly available as a deliverable of this project.

Authors' Addresses

Ben Niven-Jenkins
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK

Email: ben@velocix.com

Francois Le Faucheur
Cisco Systems
Greenside, 400 Avenue de Roumanille
Sophia Antipolis 06410
France

Phone: +33 4 97 23 26 19
Email: flefauch@cisco.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
USA

Email: nabil.bitar@verizon.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 21, 2012

K. Leung, Ed.
Cisco
Y. Lee, Ed.
Comcast

October 19, 2011

Content Distribution Network Interconnection (CDNI) Requirements
draft-ietf-cdni-requirements-01

Abstract

Content Delivery Networks (CDNs) are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. There is a requirement for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to end users. The Content Distribution Network Interconnection (CDNI) working group has been chartered to develop an interoperable and scalable solution for such CDN interconnection.

The goal of the present document is to outline the requirements for the solution and interfaces to be specified by the CDNI working group. This draft is a work in progress and requirements may be added, modified, or removed by the working group.

Requirements Language

The key words "High Priority", "Medium Priority" and "Low Priority" in this document are to be interpreted in the following way:

- "High Priority" indicates requirements that are to be supported by the CDNI interfaces. A requirement is stated as "High Priority" when it is established by the working group that it can be met without compromising the targeted schedule for WG deliverables, or when it is established that specifying a solution without meeting this requirement would not make sense and would justify re-adjusting the WG schedule, or both. This is tagged as "[HIGH]".

- "Medium Priority" indicates requirements that are to be supported by the CDNI interfaces unless the WG realizes at a later stage that attempting to meet this requirement would compromise the overall WG schedule (for example it would involve complexities that would result in significantly delaying the deliverables). This is tagged as "[MED]".

- "Low Priority" indicates requirements that are to be supported by the CDNI interfaces provided that dedicating WG resources to this work does not prevent addressing "High Priority" and "Medium Priority" requirements and that attempting to meet this requirement would not compromise the overall WG schedule. This is tagged as "[LOW]".

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future. In the face of this growth, Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for end users, and increased robustness of delivery. For these reasons CDNs are frequently used for large-scale content delivery. As a result, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs. It is generally desirable that a given content item can be delivered to an End User regardless of that End User's location or attachment network. However, the footprint of a given CDN in charge of delivering a given content may not expand close enough to the End User's current location or attachment network to realize the cost benefit and user experience that a more distributed CDN would provide. This creates a requirement for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users. The Content Distribution Network Interconnection (CDNI) working group has been chartered to develop an interoperable and scalable solution for such CDN interconnection.

[I-D.jenkins-cdni-problem-statement] outlines the problem area that the CDNI working group is chartered to address. [I-D.bertrand-cdni-use-cases] discusses the use cases for CDN Interconnection. [I-D.davie-cdni-framework] discusses the technology framework for the CDNI solution and interfaces.

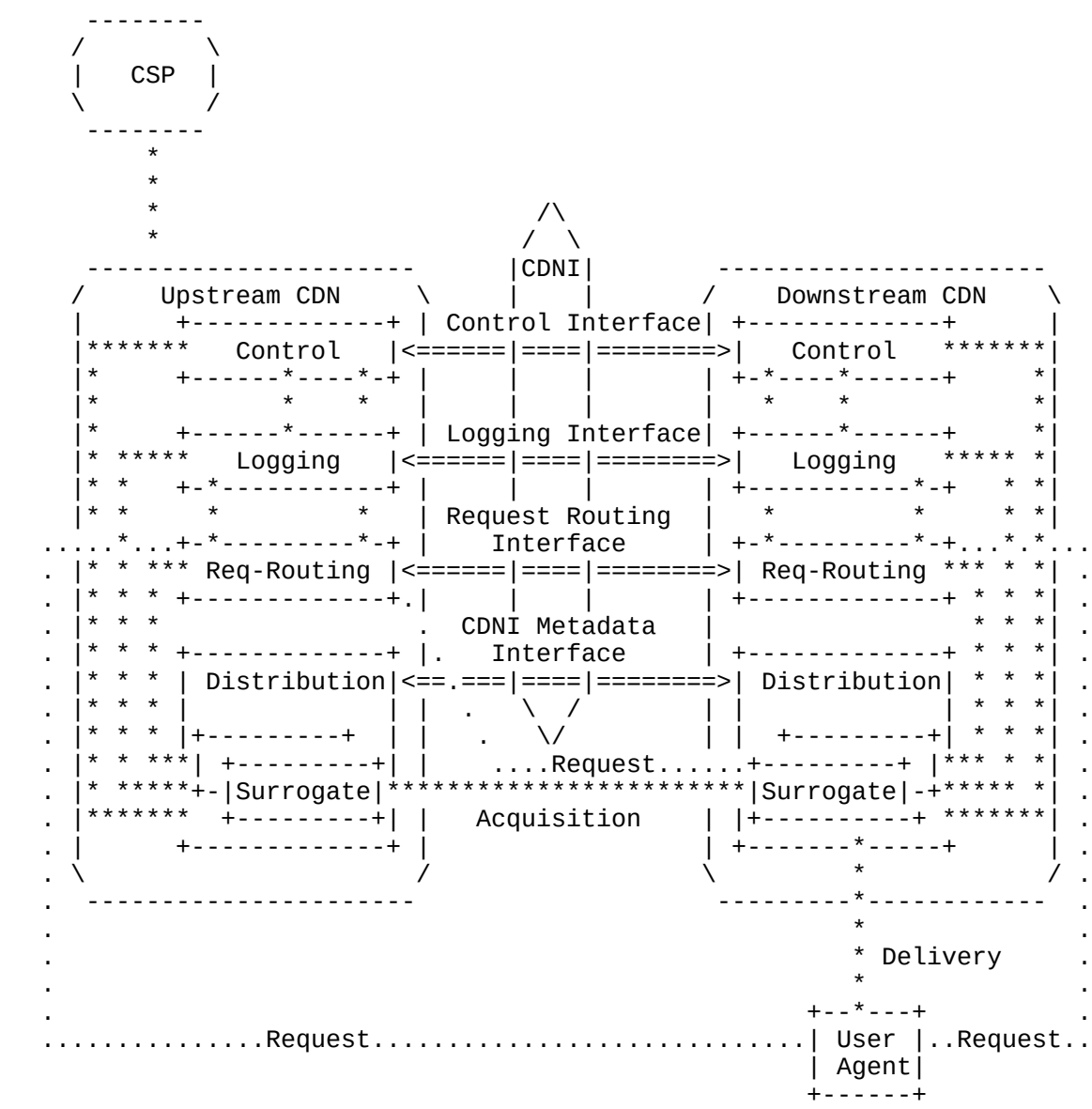
The goal of the present document is to document the requirements for the CDNI solution and interfaces. In order to meet the timelines defined in the working group charter, the present document categorizes the CDNI requirements as "High Priority", "Medium Priority", and "Low Priority".

1.1. Terminology

This document uses the terminology defined in section 1.1 of [I-D.davie-cdni-framework].

2. CDNI Model and CDNI Interfaces

For convenience Figure 1 from [I-D.davie-cdni-framework] illustrating the CDNI problem area and the CDNI protocols is replicated below.



<==> interfaces inside the scope of CDNI

**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Model and CDNI Interfaces

3. Generic CDNI Requirements

This section identifies generic requirements independent of the individual CDNI interfaces. Some of those are expected to affect multiple or all interfaces.

GEN-1 [MED] Wherever possible, the CDNI interfaces should reuse or leverage existing IETF protocols.

GEN-2 [HIGH] The CDNI solution shall not require a change, or an upgrade, to the User Agent to benefit from content delivery through interconnected CDNs.

GEN-3 [HIGH] The CDNI solution shall not require a change, or an upgrade, to the Content Service Provider to benefit from content delivery through interconnected CDNs.

GEN-4 [HIGH] The CDNI solution shall not require intra-CDN information to be exposed to other CDNs for effective and efficient delivery of the content. Examples of intra-CDN information include surrogate topology, surrogate status, cached content, etc.

GEN-5 [HIGH] The CDNI solution shall support delivery to the user agent based on HTTP [RFC2616]. (Note that while delivery and acquisition "data plane" protocols are out of the CDNI solution scope, the CDNI solution "control plane" protocols are expected to participate in enabling, selecting or facilitating operations of such acquisition and delivery protocols. Hence it is useful to state requirements on the CDNI solution in terms of which acquisition and delivery protocols).

GEN-6 [HIGH] The CDNI solution shall support acquisition across CDNs based on HTTP [RFC2616].

GEN-7 [LOW] The CDNI solution may support delivery to the user agent based on protocols other than HTTP.

GEN-8 [LOW] The CDNI solution may support acquisition across CDNs based on protocols other than HTTP.

GEN-9 [MED] The CDNI solution should support cascaded CDN redirection (CDN1 redirects to CDN2 that redirects to CDN3) to an arbitrary number of levels beyond the first level.

GEN-10 [MED] The CDNI solution should support an arbitrary topology of interconnected CDNs (i.e. the CDN topology cannot be restricted to a tree, a loop-free topology, etc.).

GEN-11 [HIGH] The CDNI solution shall prevent looping of any CDNI information exchange.

GEN-12 [HIGH] When making use of third party reference, the CDNI solution shall consider the potential issues associated with the use of various format of third-party references (e.g. NAT or IPv4/IPv6 translation potentially breaking third-party references based on an IP addresses such as URI containing IPv4 or IPv6 address litterals, split DNS situations potentially breaking third-party references based on DNS fully qualified domain names) and wherever possible avoid, minimize or mitigate the associated risks based on the specifics of the environments where the reference is used (e.g. likely or unlikely presence of NAT in the path). In particular, this applies to situations where the CDNI solution needs to construct and convey uniform resource identifiers for directing/redirecting a content request, as well as to situations where the CDNI solution needs to pass on a third party reference (e.g. to identify a User Agent) in order to allow another entity to make a more informed decision (e.g. make a more informed request routing decision by attempting to derive location information from the third party reference).

GEN-13

GEN-14 [HIGH] The CDNI solution shall support HTTP Adaptive Bit Rate (ABR) content.

4. CDNI Control Interface Requirements

The primary purpose of the CDNI Control interface is to initiate the interconnection across CDNs, bootstrap the other CDNI interfaces and trigger actions into the Downstream CDN by the Upstream CDN (such as delete object from caches or trigger pre-positioned content acquisition). We observe that while the CDNI Control interface is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols.

CNTL-1 [HIGH] The CDNI Control interface shall allow the Upstream CDN to request that the Downstream CDN (and, if cascaded CDNs are supported by the solution, that the potential cascaded Downstream CDNs) perform the following actions on an object or object set:

* Mark an object(s) and/or its CDNI metadata as "stale" and revalidate them before they are delivered again

* Delete an object(s) and/or its CDNI metadata from the CDN surrogates and any storage. Only the object(s) and CDNI metadata that pertain to the requesting Upstream CDN are allowed to be purged.

CNTL-2 [HIGH] The CDNI Control interface shall allow the downstream CDN to report on the completion of these actions (by itself, and if cascaded CDNs are supported by the solution, by potential cascaded Downstream CDNs), in a manner appropriate for the action (e.g. synchronously or asynchronously).

CNTL-3 [HIGH] The CDNI Control interface shall support initiation and control by the Upstream CDN of pre-positioned CDNI metadata acquisition by the Downstream CDN.

CNTL-4 [MED] The CDNI Control interface should support initiation and control by the Upstream CDN of pre-positioned content acquisition by the Downstream CDN.

CNTL-5 [LOW] The CDNI Control interface may allow a CDN to establish, update and terminate a CDN interconnection with another CDN whereby one CDN can act as a Downstream CDN for the other CDN (that acts as an Upstream CDN).

CNTL-6 [LOW] The CDNI Control interface may allow control of the CDNI interconnection between any two CDNs independently for each direction (i.e. For the direction where CDN1 is the Upstream CDN and CDN2 is the Downstream CDN, and for the direction where CDN2 is the Upstream CDN and CDN1 is the Downstream CDN).

CNTL-7 [LOW] The CDNI Control interface may allow bootstrapping of the Request-Routing interface. For example, this can potentially include:

* negotiation of the Request-Routing method (e.g. DNS vs HTTP, if more than one method is specified)

* discovery of the Request-Routing protocol endpoints

* information necessary to establish secure communication between the Request-Routing protocol endpoints.

CNTL-8 [LOW] The CDNI Control interface may allow bootstrapping of the CDNI Metadata interface. This information could, for example, include:

* discovery of the CDNI Metadata signaling protocol endpoints

* information necessary to establish secure communication between the CDNI Metadata signaling protocol endpoints.

CNTL-9 [LOW] The CDNI Control interface may allow bootstrapping of the Content Acquisition interface. This could, for example, include exchange and negotiation of the Content Acquisition protocols to be used across the CDNs (e.g. HTTP, HTTPS, FTP, ATIS C2).

CNTL-10 [LOW] The CDNI Control interface may allow exchange and negotiation of delivery authorization mechanisms to be supported across the CDNs (e.g. URI signature based validation).

CNTL-11 [LOW] The CDNI Control interface may allow bootstrapping of the CDNI Logging interface. This information could, for example, include:

* discovery of the Logging protocol endpoints

* information necessary to establish secure communication between the Logging protocol endpoints

* negotiation/definition of the log file format and set of fields to be exported through the Logging protocol, with some granularity (e.g. On a per content type basis).

* negotiation/definition of parameters related to transaction Logs export (e.g., export protocol, file compression, export frequency, directory).

5. CDNI Request Routing Interface Requirements

The main function of the Request Routing interface is to allow the Request-Routing systems in interconnected CDNs to communicate to

facilitate redirection of the request across CDNs.

REQ-1 [HIGH] The CDNI Request-Routing interface shall allow the Downstream CDN to communicate to the Upstream CDN coarse information about the Downstream CDN ability and/or willingness to handle requests from the Upstream CDN. For example, this could potentially include a binary signal ("Downstream CDN ready/not-ready to take additional requests from Upstream CDN") to be used in case of excessive load or failure condition in the Downstream CDN.

REQ-2 [MED] The CDNI Request-Routing interface should allow the Downstream CDN to communicate to the Upstream CDN aggregate information to facilitate CDN selection during request routing, such as Downstream CDN capabilities, resources and affinities (i.e. Preferences or cost). This information could, for example, include:

* supported content types and delivery protocols

* footprint (e.g. layer-3 coverage)

* a set of metrics/attributes (e.g. Streaming bandwidth, storage resources, distribution and delivery priority)

* a set of affinities (e.g. Preferences, indication of distribution/delivery fees)

* information to facilitate request redirection (e.g. Reachability information of Downstream CDN Request Routing system).

[Note: Some of this information - such as supported content types and delivery protocols- may also potentially be taken into account by the distribution system in the Upstream CDN for pre-positioning of content and/or metadata in the Downstream CDN in case of pre-positioned content acquisition and/or pre-positioned CDNI metadata acquisition.]

REQ-3 [MED] In the case of cascaded redirection, the CDNI Request-Routing interface shall allow the Downstream CDN to also include in the information communicated to the Upstream CDN, information on the capabilities, resources and affinities of CDNs to which the Downstream CDN may (in turn) redirect requests received by the Upstream CDN. In that case, the CDNI Request-Routing interface shall prevent looping of such information exchange.

REQ-4 [LOW] The CDNI Request-Routing interface may allow the Downstream CDN to communicate to the Upstream CDN aggregate information on CDNI administrative limits and policy. This information can be taken into account by the Upstream CDN Request Routing system in its CDN Selection decisions. This information could, for example, include:

* maximum number of requests redirected by the Upstream CDN to be served simultaneously by the Downstream CDN

* maximum aggregate volume of content (e.g. in Terabytes) to be delivered by the Downstream CDN over a time period.

REQ-5 [HIGH] The CDNI Request-Routing architecture and interface shall support efficient request-routing for small objects. This may, for example, call for a mode of operation (e.g. DNS-based request routing) where freshness and accuracy of CDN/Surrogate selection can be traded-off against reduced request-routing load (e.g. Via lighter-weight queries and caching of request-routing decisions).

REQ-6 [HIGH] The CDNI Request-Routing architecture and interface shall support efficient request-routing for large objects. This may, for example, call for a mode of operation (e.g. HTTP-based request routing) where freshness and accuracy of CDN/Surrogate selection justifies a per-request decision and a per-request CDNI Request-Routing protocol call.

REQ-7 [HIGH] The CDNI Request-Routing architecture shall support recursive CDNI request routing.

REQ-8 [HIGH] The CDNI Request-Routing architecture shall support iterative CDNI request routing.

REQ-9 [MED] In case of detection of a request redirection loop, the CDNI Request-Routing loop prevention mechanism should allow routing of the request by avoiding the loop (as opposed to the request loop being simply interrupted without routing the request).

REQ-10 [MED] The CDNI Request-Routing protocol should support a mechanism allowing enforcment of a limit on the number of successive CDN redirections for a given request.

REQ-11 [LOW] The CDNI Request-Routing protocol may support a mechanism allowing an upstream CDN to avoid redirecting a request to a downstream CDN if that is likely to result in the total redirection time exceeding some limit.

REQ-12 [HIGH] The CDNI Request-Routing protocol shall allow the Upstream CDN to include, in the query to the Downstream CDN, the necessary information to allow the Downstream CDN to process the redirection query. This could, for example, include:

* information from which the location of the user-agent that originated the request can be inferred (e.g. User Agent fully qualified domain name in case of HTTP-based Request Routing, DNS Proxy fully qualified domain name in case of DNS-based Request Routing)

* requested resource information (e.g. Resource URI in case of HTTP-based Request Routing, Resource hostname in case of DNS-based Request Routing)

* additional available request information (e.g. request headers in case of HTTP-based Request Routing).

REQ-13 [LOW] The CDNI Request-Routing protocol may also allow the Upstream CDN to convey information pointing to CDNI metadata applicable (individually or through inheritance) to the requested content. For illustration, the CDNI metadata pointed to could potentially include metadata that is applicable to any content, metadata that is applicable to a content collection (to which the requested content belongs) and/or metadata that is applicable individually to the requested content.

REQ-14 [HIGH] The CDNI Request-Routing interface shall allow the Downstream CDN to include the following information in the response to the Upstream CDN:

* status code, in particular indicating acceptance or rejection of request (e.g. Because the Downstream CDN is unwilling or unable to serve the request). In case of rejection, an error code is also to be provided, which allows the Upstream CDN to react appropriately (e.g. Select another Downstream CDN, or serve the request itself)

* redirection information (e.g. Resource URI in case of HTTP-based Request Routing, equivalent of a DNS record in case of DNS-based Request Routing).

6. CDNI Metadata Distribution Interface Requirements

The primary function of the CDNI Metadata Distribution interface is to allow the Distribution system in interconnected CDNs to communicate to ensure Content Distribution Metadata with inter-CDN scope can be exchanged across CDNs. We observe that while the CDNI Metadata Distribution protocol is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols. For example, a subset of the CDNI metadata might be conveyed in-band along with the actual content acquisition across CDNs (e.g. content MD5 in HTTP header) while another subset might require an out-of-band interface & protocol (e.g. geo-blocking information).

META-1 [HIGH] The CDNI Metadata Distribution interface shall allow the Upstream CDN to provide the Downstream CDN with content distribution metadata of inter-CDN scope.

META-2 [HIGH] The CDNI Metadata Distribution interface shall support exchange of CDNI metadata for both the dynamic content acquisition model and the pre-positioning content acquisition model.

META-3 [HIGH] The CDNI Metadata Distribution interface shall support a mode where no, or a subset of, the Metadata is initially communicated to the Downstream CDN along with information about how/where to acquire the rest of the CDNI Metadata (i.e. Dynamic CDNI metadata acquisition).

META-4 [MED] The CDNI Metadata Distribution interface should support a mode where all the relevant Metadata is initially communicated to the Downstream CDN (i.e. Pre-positioned CDNI metadata acquisition).

META-5 [HIGH] Whether in the pre-positioned content acquisition model or in the dynamic content acquisition model, the CDNI Metadata Distribution interface shall provide the necessary information to allow the Downstream CDN to acquire the content from an upstream source (e.g. Acquisition protocol and Uniform Resource Identifier in Upstream CDN- or rules to construct this URI).

META-6 [HIGH] The CDNI metadata shall allow signaling of one or more upstream sources, where each upstream source can be in the Upstream CDN, in another CDN, the CSP origin server or any arbitrary source designated by the Upstream CDN. Note that some upstream sources (e.g. the content origin server)

may or may not be willing to serve the content to the Downstream CDN, if this policy is known to the upstream CDN then it may omit those sources when exchanging CDNI metadata.

META-7 [HIGH] The CDNI Metadata Distribution interface shall allow the Upstream CDN to request addition and modification of CDNI Metadata into the Downstream CDN.

META-8 [HIGH] The CDNI Metadata Distribution interface shall allow removal of obsolete CDNI Metadata from the Downstream CDN (this could, for example, be achieved via an explicit removal request from the Upstream CDN or via expiration of a Time-To-Live associated to the Metadata).

META-9 [HIGH] The CDNI Metadata Distribution interface shall allow association of CDNI Metadata at the granularity of individual object. This is necessary to achieve fine-grain Metadata distribution at the level of an individual object when necessary.

META-10 [HIGH] The CDNI Metadata Distribution interface shall allow association of CDNI Metadata at the granularity of an object set. This is necessary to achieve scalable distribution of metadata when a large number of objects share the same distribution policy.

META-11 [HIGH] The CDNI Metadata Distribution interface shall support multiple levels of inheritance with precedence to more specific metadata. For example, the CDNI Metadata Distribution protocol may support metadata that is applicable to any content, metadata that is applicable to a content collection and metadata that is applicable to an individual content where content level metadata overrides content collection metadata that overrides metadata for any content.

META-12 [HIGH] The CDNI Metadata Distribution interface shall ensure that conflicting metadata with overlapping scope are prevented or deterministically handled.

META-13 [HIGH] The CDNI Metadata Distribution interface shall provide indication by the Downstream CDN to the Upstream CDN of whether the CDNI metadata (and corresponding future request redirections) is accepted or rejected. When rejected, the CDNI Metadata Distribution protocol Must allow the Downstream CDN to provide information about the cause of the rejection.

META-14 [HIGH] The CDNI Metadata Distribution interface shall allow signaling of content distribution control policies. For example, this could potentially include:

* geo-blocking information (i.e. Information defining geographical areas where the content is to be made available or blocked)

* availability windows (i.e. Information defining time windows during which the content is to be made available or blocked; expiration time may also be included to remove content)

* delegation whitelist/blacklist (i.e. Information defining which downstream CDNs the content may/may not be delivered through)

META-15 [HIGH] The CDNI Metadata interface shall be able to exchange a set of well-accepted metadata elements with specified semantics (e.g. start of time window, end of time window).

META-16 [HIGH] The CDNI Metadata interface shall allow exchange of opaque metadata element, whose semantic is not defined in CDNI but established by private CDN agreement.

META-17 [HIGH] The CDNI Metadata Distribution interface shall allow signaling of authorization checks and validation that are to be performed by the surrogate before delivery. For example, this could potentially include:

* need to validate URI signed information (e.g. Expiry time, Client IP address).

META-18 [LOW] The CDNI Metadata Distribution interface may allow signaling of CDNI-relevant surrogate cache behavior parameters. For example, this could potentially include:

* control of whether the query string of HTTP URI is to be ignored by surrogate cache

* content revalidation parameters (e.g. TTL)

7. CDNI Logging Interface Requirements

This section identifies the requirements related to the CDNI Logging interface. We observe that while the CDNI Logging interface is currently discussed as a single "protocol", further analysis will

determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols.

LOG-1 [HIGH] The CDNI logging architecture and interface shall ensure reliable logging of CDNI events.

LOG-2 [HIGH] The CDNI Logging interface shall provide logging of deliveries to User Agents performed by the Downstream CDN as a result of request redirection by the Upstream CDN.

LOG-3 [MED] In the case of cascaded CDNs, the CDNI Logging interface shall allow the Downstream CDN to report to the Upstream CDN logging for deliveries performed by the Downstream CDN itself as well as logging for deliveries performed by cascaded CDNs on behalf of the Downstream CDN.

LOG-4 [HIGH] The CDNI Logging interface shall provide logging of distribution performed by the Upstream CDN as a result of acquisition request by the Downstream CDN.

LOG-5 [HIGH] The CDNI Logging interface shall support batch/offline exchange of logging records.

LOG-6 [MED] The CDNI Logging interface should also support additional timing constraints for some types of logging records (e.g. near-real time for monitoring and analytics applications)

LOG-7 [HIGH] The CDNI Logging interface shall define a log file format and a set of fields to be exported through the Logging protocol, with some granularity (e.g. On a per content type basis).

LOG-8 [HIGH] The CDNI Logging interface shall define a transport mechanisms to exchange CDNI Logging files.

LOG-9 [LOW] The CDNI Logging interface may allow a CDN to query another CDN for relevant current logging records (e.g. For on-demand access to real-time logging information).

LOG-10 [LOW] The CDNI Logging interface may support aggregate/ summarized logs (e.g. total bytes delivered for a content regardless of individual User Agents to which it was delivered).

LOG-11 [LOW] The CDNI Logging interface may provide "quality" metrics in the logging of deliveries to User Agents performed by the Downstream CDN.

8. CDNI Security Requirements

This section identifies the requirements related to the CDNI security. Some of those are expected to affect multiple or all protocols.

SEC-1 [HIGH] All the CDNI interface shall support secure operation over unsecured IP connectivity (e.g. The Internet). This includes authentication, confidentiality, integrity protection as well as protection against spoofing and replay.

SEC-2 [HIGH] The CDNI solution shall provide sufficient protection against Denial of Service attacks. This includes protection against spoofed delivery requests sent by user agents directly to a Downstream CDN attempting to appear as if they had been redirected by a given Upstream CDN when they have not.

SEC-3 [MED] The CDNI solution should be able to ensure that for any given request redirected to a Downstream CDN, the chain of CDN Delegation (leading to that request being served by that CDN) can be established with non-repudiation.

SEC-4 [MED] The CDNI solution should be able to ensure that the Downstream CDN cannot spoof a transaction log attempting to appear as if it corresponds to a request redirected by a given Upstream CDN when that request has not been redirected by this Upstream CDN. This ensures non-repudiation by the Upstream CDN of transaction logs generated by the Downstream CDN for deliveries performed by the Downstream CDN on behalf of the Upstream CDN.

SEC-5 [LOW] The CDNI solution may provide a mechanism allowing an Upstream CDN that has credentials to acquire content from the CSP origin server (or another CDN), to allow establishment of credentials authorizing the Downstream CDN to acquire the content from the CSP origin server (or the other CDN) (e.g. In case the content cannot be acquired from the Upstream CDN).

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

This document discusses CDNI security requirements in Section 8.

11. Authors

This document reflects the contributions from the following authors:

Francois Le Faucheur

Cisco Systems

flefauch@cisco.com

Mahesh Viveganandhan

Cisco Systems

mvittal@cisco.com

Grant Watson

BT

grant.watson@bt.com

12. Acknowledgements

This document leverages the earlier work of the IETF CDI working group in particular as documented in [I-D.cain-request-routing-req], [I-D.amini-cdi-distribution-reqs] and [I-D.gilletti-cdnp-aaa-reqs].

The authors would like to thank Gilles Bertrand, Christophe Caillet, Bruce Davie, Phil Eardly, Ben Niven-Jenkins, Agustin Schapira, Emile Stephan, Eric Burger, Susan He, Kevin Ma, and Daryl Malas for their input.

13. References

13.1. Normative References

[I-D.bertrand-cdni-use-cases]
Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-bertrand-cdni-use-cases-02 (work in progress), July 2011.

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[I-D.jenkins-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-jenkins-cdni-problem-statement-02 (work in progress), March 2011.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

13.2. Informative References

[I-D.amini-cdi-distribution-reqs]
Amini, L., "Distribution Requirements for Content Internetworking", draft-amini-cdi-distribution-reqs-02 (work in progress), November 2001.

[I-D.cain-request-routing-req]
Cain, B., "Request Routing Requirements for Content Internetworking", draft-cain-request-routing-req-03 (work in progress), November 2001.

[I-D.gilletti-cdnp-aaa-reqs]
"CDI AAA Requirements, draft-gilletti-cdnp-aaa-reqs-01.txt", June 2001.

Authors' Addresses

Kent Leung (editor)
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
U.S.A.

Phone: +1 408 526 5030
Email: kleung@cisco.com

Yiu Lee (editor)
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: yiu_lee@cable.comcast.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 3, 2014

K. Leung, Ed.
Cisco
Y. Lee, Ed.
Comcast
Jan 30, 2014

Content Distribution Network Interconnection (CDNI) Requirements
draft-ietf-cdni-requirements-17

Abstract

Content delivery is frequently provided by specifically architected and provisioned Content Delivery Networks (CDNs). As a result of significant growth in content delivered over IP networks, existing CDN providers are scaling up their infrastructure. Many Network Service Providers and Enterprise Service Providers are also deploying their own CDNs. To deliver contents from the Content Service Provider (CSP) to end users, the contents may traverse across multiple CDNs. This creates a need for interconnecting (previously) standalone CDNs so that they can collectively act as a single delivery platform from the CSP to the end users.

The goal of the present document is to outline the requirements for the solution and interfaces to be specified by the CDNI working group.

Table of Contents

1. Introduction

The volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future. In the face of this growth, Content Delivery Networks (CDNs) provide numerous benefits: reduced delivery cost for cacheable content, improved quality of experience for end users, and increased robustness of delivery. For these reasons CDNs are frequently used for large-scale content delivery. As a result of the significant growth in content delivered over IP networks, existing CDN providers are scaling up their infrastructure and many Network Service Providers and Enterprise Service Providers are deploying their own CDNs. Subject to the policy of the Content Service Provider (CSP), it is generally desirable that a given item of content can be delivered to an end user regardless of that end user's location or attachment network. This creates a need for interconnecting (previously) standalone CDNs so they can interoperate and collectively behave as a single delivery infrastructure. The Content Distribution Network Interconnection (CDNI) working group has been chartered to develop an interoperable and scalable solution for such CDN interconnections.

CDNI Problem Statement [RFC6707] outlines the problem area that the CDNI working group is chartered to address. Use Cases for CDNI [RFC6770] discusses the use cases for CDN Interconnection. Framework for CDN Interconnection [I-D.ietf-cdni-framework] discusses the technology framework for the CDNI solution and interfaces.

The goal of the present document is to document the requirements for the CDNI solution and interfaces. In order to meet the timelines defined in the working group charter, the present document categorizes the CDNI requirements as "High Priority", "Medium Priority", and "Low Priority".

1.1. Terminology

This document uses the terminology defined in [RFC6707]. In addition, the key words "High Priority", "Medium Priority" and "Low Priority" in this document are to be interpreted in the following way:

- "High Priority": When a requirement is tagged as "{HIGH}", it is considered by the working group as an essential function for CDNI and necessary to a deployable solution. This requirement has to be met even if it causes a delay in the delivery by the working group of a deployable solution.

- "Medium Priority": When a requirement is tagged as "{MED}", it is considered by the working group as an important function for CDNI. This requirement has to be met, unless it is established that attempting to meet this requirement would cause a delay in the delivery by the working group of a deployable solution.

- "Low Priority": When a requirement is tagged as "{LOW}", it is considered by the working group as a useful function for CDNI. The working group will attempt to meet this requirement as long as it does not prevent meeting the "High Priority" and "Medium Priority" requirements and does not cause a delay in the delivery by the working group of a deployable solution.

2. CDNI Model and CDNI Interfaces

The "CDNI Expanded Model and CDNI Interfaces" figure and brief descriptions of the CDNI interfaces in [I-D.ietf-cdni-framework] are replicated below for convenience. That document contains the definitive reference model and descriptions for the CDNI interfaces.

- CDNI Control interface (CI): Operations to bootstrap and parameterize the other CDNI interfaces, as well as operations to pre-position, revalidate, and purge both metadata and content. The latter subset of operations is sometimes collectively called the "Trigger interface."

- CDNI Request Routing interface: Operations to determine what CDN (and optionally what surrogate within a CDN) is to serve end-user’s requests. This interface is actually a logical bundling of two separate but related interfaces:

  * CDNI Footprint & Capabilities Advertisement interface (FCI): Asynchronous operations (as defined in [I-D.ietf-cdni-framework]) to exchange routing information (e.g., the network footprint and capabilities served by a given CDN) that enables CDN selection for subsequent user requests; and

  * CDNI Request Routing Redirection interface (RI): Synchronous operations (as defined in [I-D.ietf-cdni-framework]) to select a delivery CDN (surrogate) for a given user request.

- CDNI Metadata interface (MI): Operations to communicate metadata that governs how the content is delivered by interconnected CDNs. Examples of CDNI metadata include geo-blocking directives, availability windows, access control mechanisms, and purge directives. It may include a combination of:

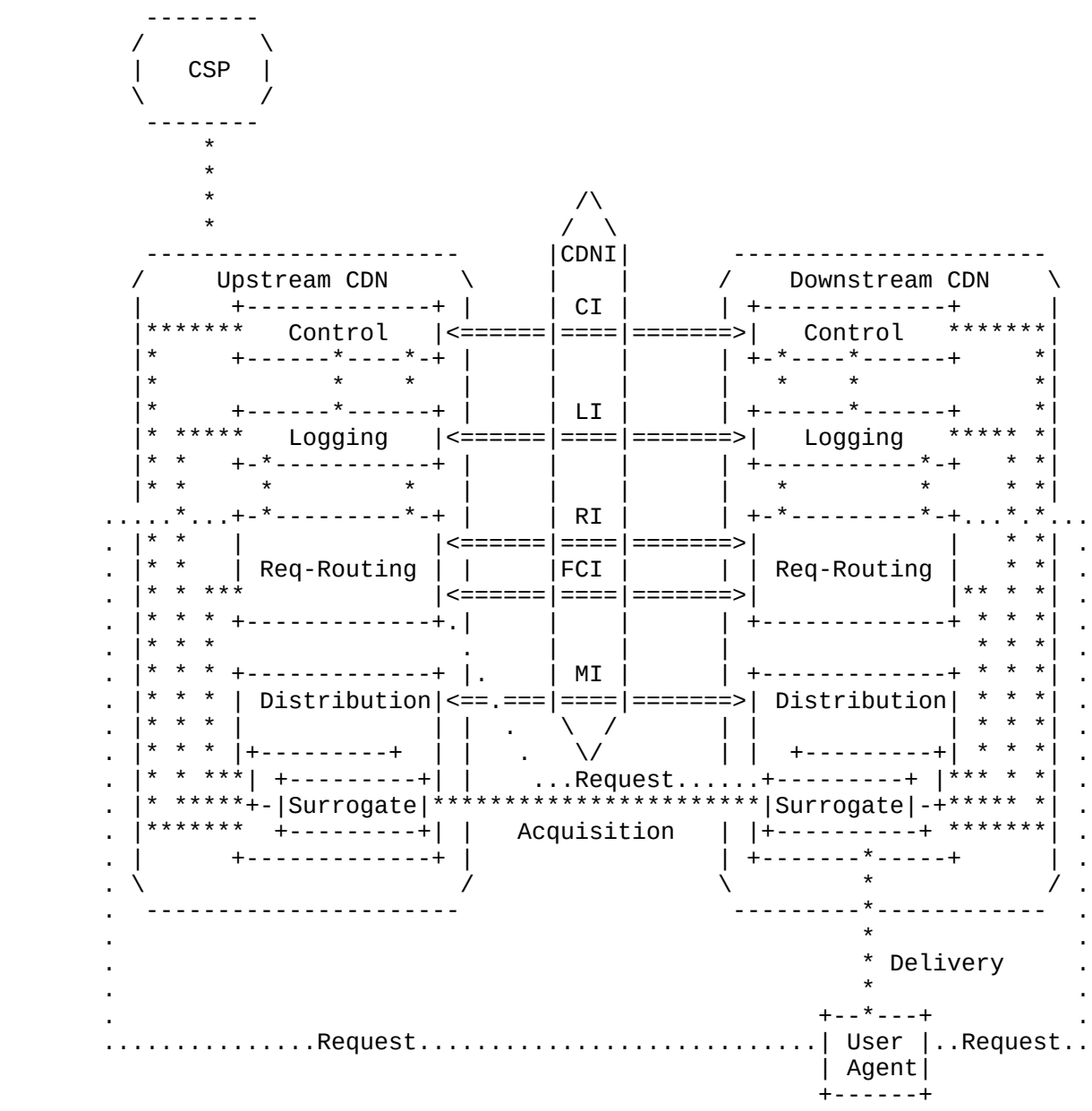
  * Asynchronous operations to exchange metadata that govern subsequent user requests for content; and

  * Synchronous operations that govern behavior for a given user request for content.

- CDNI Logging interface (LI): Operations that allow interconnected CDNs to exchange relevant activity logs. It may include a combination of:

  * Real-time exchanges, suitable for runtime traffic monitoring; and

  * Offline exchanges, suitable for analytics and billing.



<==> interfaces inside the scope of CDNI

**** and .... interfaces outside the scope of CDNI

Figure 1: CDNI Expanded Model and CDNI Interfaces

3. Generic CDNI Requirements

This section identifies generic requirements independent of the individual CDNI interfaces. Some of those are expected to affect multiple or all interfaces. Management is an important aspect of CDN operation. The fault and performance management is covered in CDNI Logging interface requirements. The other types of management are specific to the CDN provider and not needed for interoperability between CDN providers.

GEN-1 {MED} Wherever possible, the CDNI interfaces should reuse or leverage existing IETF protocols.

GEN-2 {HIGH} The CDNI solution shall not require a change, or an upgrade, to the User Agent to benefit from content delivery through interconnected CDNs.

GEN-3 {HIGH} The CDNI solution shall not require a change, or an upgrade, to the Content Service Provider delivering content through a single CDN, to benefit from content delivery through interconnected CDNs.

GEN-4 {HIGH} The CDNI solution shall not depend on intra-CDN information to be exposed to other CDNs for effective and efficient delivery of the content. Examples of intra-CDN information include surrogate topology, surrogate status, cached content, etc.

GEN-5 {HIGH} The CDNI solution shall support CDN interconnection when delivery to the User Agent is based on HTTP [RFC2616]. (Note that while delivery and acquisition "data plane" protocols are out of the CDNI solution scope, the CDNI solution "control plane" protocols are expected to participate in enabling, selecting or facilitating operations of such acquisition and delivery protocols. Hence it is useful to state requirements on the CDNI solution in terms of specifying which acquisition and delivery protocols are to be supported).

GEN-6 {HIGH} The CDNI solution shall support acquisition across CDNs based on HTTP [RFC2616]. (The note above applies to this requirement too)

GEN-7 {LOW} The CDNI solution may support delivery to the User Agent based on protocols other than HTTP.

GEN-8 {LOW} The CDNI solution may support acquisition across CDNs based on protocols other than HTTP.

GEN-9 {MED} The CDNI solution should support cascaded CDN redirection (CDN1 redirects to CDN2 that redirects to CDN3) to an arbitrary number of levels beyond the first level.

GEN-10 {MED} The CDNI solution should support an arbitrary topology of interconnected CDNs (i.e. the topology of interconnected CDNs cannot be restricted to a tree, ring, star, etc.).

GEN-11 {HIGH} The CDNI solution shall prevent looping of any CDNI information exchange.

GEN-12 {HIGH} When making use of third party reference, the CDNI solution shall consider the potential issues associated with the use of various format of third-party references (e.g. NAT or IPv4/IPv6 translation potentially breaking third-party references based on an IP addresses such as URI containing IPv4 or IPv6 address literals, split DNS situations potentially breaking third-party references based on DNS fully qualified domain names) and wherever possible avoid, minimize or mitigate the associated risks based on the specifics of the environments where the reference is used (e.g. likely or unlikely presence of NAT in the path). In particular, this applies to situations where the CDNI solution needs to construct and convey uniform resource identifiers for directing/redirecting a content request, as well as to situations where the CDNI solution needs to pass on a third party reference (e.g. identify the IP address of a User Agent) in order to allow another entity to make a more informed decision (e.g. make a more informed request routing decision by attempting to derive location information from the third party reference).

GEN-13 {HIGH} The CDNI solution shall support HTTP Adaptive Streaming content.

4. CDNI Control Interface Requirements

The primary purpose of the CDNI Control interface (CI) is to initiate the interconnection across CDNs, bootstrap the other CDNI interfaces and trigger actions into the Downstream CDN by the Upstream CDN (such as delete object from caches or trigger pre-positioned content acquisition). The working group attempts to align requirements with the appropriate interface; however, solutions to these requirements may apply to a different interface or another interface in addition

to the interface it is associated with.

CI-1 {HIGH} The CDNI Control interface shall allow the Upstream CDN to request that the Downstream CDN, including downstream cascaded CDNs, delete an object or set of objects and/or its CDNI metadata from the CDN surrogates and any storage. Only the object(s) and CDNI metadata that pertain to the requesting Upstream CDN are allowed to be purged.

CI-2 {MED} The CDNI Control interface should allow for multiple content items identified by a Content Collection ID to be purged using a single Content Purge action.

CI-3 {MED} The CDNI Control interface should allow the Upstream CDN to request that the Downstream CDN, including downstream cascaded CDNs, mark an object or set of objects and/or its CDNI metadata as "stale" and revalidate them before they are delivered again.

CI-4 {HIGH} The CDNI Control interface shall allow the Downstream CDN to report on the completion of these actions (by itself, and including downstream cascaded CDNs), in a manner appropriate for the action (e.g. synchronously or asynchronously). The confirmation receipt should include a success or failure indication. The failure indication and the reason are included if the Downstream CDN cannot delete the content in its storage.

CI-5 {MED} The CDNI Control interface should support initiation and control by the Upstream CDN of pre-positioned CDNI metadata acquisition by the Downstream CDN.

CI-6 {MED} The CDNI Control interface should support initiation and control by the Upstream CDN of pre-positioned content acquisition by the Downstream CDN.

CI-7 {LOW} The CDNI Control interface may allow a CDN to establish, update and terminate a CDN interconnection with another CDN whereby one CDN can act as a Downstream CDN for the other CDN (that acts as an Upstream CDN).

CI-8 {LOW} The CDNI Control interface may allow control of the CDNI interfaces between any two CDNs independently for each direction (e.g. For the direction where CDN1 is the Upstream CDN and CDN2 is the Downstream CDN, and for the direction where CDN2 is the Upstream CDN and CDN1 is the Downstream CDN).

CI-9 {LOW} The CDNI Control interface may allow bootstrapping of the CDNI Request Routing interface. For example, this can potentially include:

* negotiation of the request routing method (e.g. DNS vs HTTP, if more than one method is specified)

* discovery of the CDNI Request Routing interface endpoints

* information necessary to establish secure communication between the CDNI Request Routing interface endpoints.

CI-10 {LOW} The CDNI Control interface may allow bootstrapping of the CDNI Metadata interface. This information could, for example, include:

* discovery of the CDNI Metadata interface endpoints

* information necessary to establish secure communication between the CDNI Metadata interface endpoints.

CI-11 {LOW} The CDNI Control interface may allow bootstrapping of the Content Acquisition interface. This could, for example, include exchange and negotiation of the Content Acquisition methods to be used across the CDNs (e.g. HTTP, HTTPS, FTP, ATIS C2[ATIS-0800042]).

CI-12 {LOW} The CDNI Control interface may allow bootstrapping of the CDNI Logging interface. This information could, for example, include:

* discovery of the CDNI Logging interface endpoints

* information necessary to establish secure communication between the CDNI Logging interface endpoints

* negotiation/definition of the log file format and set of fields to be exported through the logging protocol, with some granularity (e.g. On a per content type basis).

* negotiation/definition of parameters related to transaction logs export (e.g., export protocol, file compression, export frequency, directory).

5. CDNI Request Routing Redirection Interface Requirements

The main function of the CDNI Request Routing Redirection interface (RI) is to allow the Request-Routing systems in interconnected CDNs to communicate to facilitate redirection of the request across CDNs.

RI-1 {HIGH} The CDNI Request Routing Redirection interface shall support efficient request routing for small objects. This may, for example, call for a mode of operation (e.g. DNS-based request routing) where freshness and accuracy of CDN/ Surrogate selection can be traded-off against reduced request routing load (e.g. Via lighter-weight queries and caching of request routing decisions).

RI-2 {HIGH} The CDNI Request Routing Redirection interface shall support efficient request routing for large objects. This may, for example, call for a mode of operation (e.g. HTTP-based request routing) where freshness and accuracy of CDN/ Surrogate selection justifies a per-request decision and a per-request CDNI Request-Routing protocol call.

RI-3 {HIGH} The CDNI Request Routing Redirection interface shall support recursive CDNI request routing.

RI-4 {HIGH} The CDNI Request Routing Redirection interface shall support iterative CDNI request routing.

RI-5 {MED} In case of detection of a request redirection loop, the CDNI Request Routing Redirection Interface’s loop prevention mechanism should allow redirection of the request on an alternate CDN path (as opposed to the request not being redirected at all).

RI-6 {MED} The CDNI Request Routing Redirection interface should support a mechanism allowing enforcement of a limit on the number of successive CDN redirections for a given request.

RI-7 {LOW} The CDNI Request Routing Redirection interface may support a mechanism allowing an Upstream CDN to avoid redirecting a request to a Downstream CDN if that is likely to result in the total redirection time exceeding some limit.

RI-8 {HIGH} The CDNI Request Routing Redirection interface shall allow the Upstream CDN to include, in the query to the Downstream CDN, the necessary information to allow the Downstream CDN to process the redirection query. This could, for example, include:

* information from which the geographic region pertaining to the IP address of the User Agent that originated the request can be inferred (e.g. User Agent fully qualified domain name in case of HTTP-based Request Routing, DNS Proxy fully qualified domain name in case of DNS-based Request Routing)

* requested resource information (e.g. Resource URI in case of HTTP-based Request Routing, Resource hostname in case of DNS-based Request Routing)

* additional available request information (e.g. request headers in case of HTTP-based Request Routing).

RI-9 {LOW} The CDNI Request Routing Redirection interface may also allow the Upstream CDN to convey information pointing to CDNI metadata applicable (individually or through inheritance) to the requested content. For illustration, the CDNI metadata pointed to could potentially include metadata that is applicable to any content, metadata that is applicable to a content collection (to which the requested content belongs) and/or metadata that is applicable individually to the requested content.

RI-10 {HIGH} The CDNI Request Routing Redirection interface shall allow the Downstream CDN to include the following information in the response to the Upstream CDN:

* status code, in particular indicating acceptance or rejection of request (e.g. Because the Downstream CDN is unwilling or unable to serve the request). In case of rejection, an error code is also to be provided, which allows the Upstream CDN to react appropriately (e.g. Select another Downstream CDN, or serve the request itself)

* redirection information (e.g. Resource URI in case of HTTP-based Request Routing, equivalent of a DNS record in case of DNS-based Request Routing).

RI-11 {HIGH} The CDNI Request Routing Redirection interface shall allow for per-chunk request routing of HTTP Adaptive Streaming content.

RI-12 {LOW} The CDNI Request Routing Redirection interface may allow the Upstream CDN to use the information conveyed by the Downstream CDN during the Recursive Request Routing process to rewrite an HTTP Adaptive Streaming manifest file.

RI-13 {LOW} The CDNI Request-Routing interface may allow the Upstream CDN to re-compute the message digest or digital signature over the invariant portion of the chunk URIs embedded in the HTTP Adaptive Streaming manifest file.

RI-14 {MED} The CDNI Request Routing Redirection interface should correlate the HTTP Adaptive Stream manifest file to the related chunks referenced in the manifest file.

RI-15 {MED} The CDNI Request Routing Redirection interface should allow for an efficient method of transferring request routing information for multiple chunks from the Downstream CDN to the Upstream CDN as part of the recursive request routing process.

6. CDNI Footprint & Capabilities Advertisement Interface Requirements

The main function of the CDNI Footprint & Capabilities Advertisement interface (FCI) is to allow the Downstream CDN to advertise the information regarding its footprint and capabilities to the Upstream CDN.

FCI-1 {HIGH} The CDNI Footprint & Capabilities Advertisement interface shall allow the Downstream CDN to communicate to the Upstream CDN coarse information about the Downstream CDN ability and/or willingness to handle requests from the Upstream CDN. For example, this could potentially include a binary signal ("Downstream CDN ready/not-ready to take additional requests from Upstream CDN") to be used in case of excessive load or failure condition in the Downstream CDN.

FCI-2 {MED} The CDNI Footprint & Capabilities Advertisement interface should allow the Downstream CDN to communicate to the Upstream CDN aggregate information to facilitate CDN selection during request routing, such as Downstream CDN capabilities, resources and affinities (i.e. Preferences or cost). This information could, for example, include:

* supported content types and delivery protocols

* footprint (e.g. layer-3 coverage)

* a set of metrics/attributes (e.g. Streaming bandwidth, storage resources, distribution and delivery priority)

* a set of affinities (e.g. Preferences, indication of distribution/delivery fees)

* information to facilitate request redirection (e.g. Reachability information of Downstream CDN Request Routing system).

[Note: Some of this information - such as supported content types and delivery protocols- may also potentially be taken into account by the distribution system in the Upstream CDN for pre-positioning of content and/or metadata in the Downstream CDN in case of pre-positioned content acquisition and/or pre-positioned CDNI metadata acquisition.]

FCI-3 {MED} In the case of cascaded redirection, the CDNI Footprint & Capabilities Advertisement interface should allow the Downstream CDN to also include in the information communicated to the Upstream CDN, information on the capabilities, resources and affinities of CDNs to which the Downstream CDN may (in turn) redirect requests received by the Upstream CDN. In that case, the CDNI Request-Routing interface shall prevent looping of such information exchange.

FCI-4 {LOW} The CDNI Footprint & Capabilities Advertisement interface may allow the Downstream CDN to communicate to the Upstream CDN aggregate information on CDNI administrative limits and policy. This information can be taken into account by the Upstream CDN Request Routing system in its CDN Selection decisions. This information could, for example, include:

* maximum number of requests redirected by the Upstream CDN to be served simultaneously by the Downstream CDN

* maximum aggregate volume of content (e.g. in Terabytes) to be delivered by the Downstream CDN over a time period.

FCI-5 {MED} The CDNI Footprint & Capabilities Advertisement interface should support advertisement of the following types of capabilities:

* delivery protocol (e.g., HTTP vs. RTMP)

* acquisition protocol (for acquiring content from an Upstream CDN)

* redirection mode (e.g., DNS Redirection vs. HTTP Redirection)

* capabilities related to CDNI Logging (e.g., supported logging mechanisms)

* capabilities related to CDNI Metadata (e.g., authorization algorithms or support for proprietary vendor metadata)

FCI-6 {LOW} The CDNI Control interface may allow exchange and negotiation of delivery authorization mechanisms to be supported across the CDNs (e.g. URI signature based validation).

FCI-7 {HIGH} The CDNI Footprint & Capabilities Advertisement interface shall support extensible fields used to convey the CDN capabilities and methods to indicate the footprint in the advertisement from the Downstream CDN to the Upstream CDN.

7. CDNI Metadata Interface Requirements

The primary function of the CDNI Metadata interface (MI) is to allow the Distribution system in interconnected CDNs to communicate to ensure Content Distribution Metadata with inter-CDN scope can be exchanged across CDNs. We observe that while the CDNI Metadata Distribution protocol is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols. For example, a subset of the CDNI metadata might be conveyed in-band along with the actual content acquisition across CDNs (e.g. content MD5 in HTTP header) while another subset might require an out-of-band interface & protocol (e.g. geo-blocking information).

MI-1 {HIGH} The CDNI Metadata interface shall allow the Upstream CDN to provide the Downstream CDN with content distribution metadata of inter-CDN scope.

MI-2 {HIGH} The CDNI Metadata interface shall support exchange of CDNI metadata for both the dynamic content acquisition model and the pre-positioning content acquisition model.

MI-3 {HIGH} The CDNI Metadata interface shall support a mode where no, or a subset of, the Metadata is initially communicated to the Downstream CDN along with information about how/where to acquire the rest of the CDNI Metadata (i.e. Dynamic CDNI metadata acquisition).

MI-4 {MED} The CDNI Metadata interface should support a mode where all the relevant Metadata is initially communicated to the Downstream CDN (i.e. Pre-positioned CDNI metadata acquisition).

MI-5 {HIGH} Whether in the pre-positioned content acquisition model or in the dynamic content acquisition model, the CDNI Metadata interface shall provide the necessary information to allow the Downstream CDN to acquire the content from an upstream source (e.g. Acquisition protocol and Uniform Resource Identifier in Upstream CDN- or rules to construct this URI).

MI-6 {HIGH} The CDNI metadata shall allow signaling of one or more upstream sources, where each upstream source can be in the Upstream CDN, in another CDN, the CSP origin server or any arbitrary source designated by the Upstream CDN. Note that some upstream sources (e.g. the content origin server) may or may not be willing to serve the content to the Downstream CDN, if this policy is known to the Upstream CDN then it may omit those sources when exchanging CDNI metadata.

MI-7 {HIGH} The CDNI Metadata interface (possibly in conjunction with the CDNI Control interface) shall allow the Upstream CDN to request addition and modification of CDNI Metadata into the Downstream CDN.

MI-8 {HIGH} The CDNI Metadata interface (possibly in conjunction with the CDNI Control interface) shall allow removal of obsolete CDNI Metadata from the Downstream CDN (this could, for example, be achieved via an explicit removal request from the Upstream CDN or via expiration of a Time-To-Live associated to the Metadata).

MI-9 {HIGH} The CDNI Metadata interface shall allow association of CDNI Metadata at the granularity of individual object. This is necessary to achieve fine-grain Metadata distribution at the level of an individual object when necessary.

MI-10 {HIGH} The CDNI Metadata interface shall allow association of CDNI Metadata at the granularity of an object set. This is necessary to achieve scalable distribution of metadata when a large number of objects share the same distribution policy.

MI-11 {HIGH} The CDNI Metadata interface shall support multiple levels of inheritance with precedence to more specific metadata. For example, the CDNI Metadata Distribution protocol may support metadata that is applicable to any content, metadata that is applicable to a content collection and metadata that is applicable to an individual content where content level metadata overrides content collection metadata that overrides metadata for any content.

MI-12 {HIGH} The CDNI Metadata interface shall ensure that conflicting metadata with overlapping scope are prevented or deterministically handled.

MI-13 {HIGH} The CDNI Metadata interface shall allow signaling of content distribution control policies. For example, this could potentially include:

* geo-blocking information (i.e. Information defining geographical areas where the content is to be made available or blocked)

* availability windows (i.e. Information defining time windows during which the content is to be made available or blocked; expiration time may also be included to remove content)

* delegation whitelist/blacklist (i.e. Information defining which Downstream CDNs the content may/may not be delivered through)

MI-14 {HIGH} The CDNI Metadata interface shall be able to exchange a set of metadata elements with specified semantics (e.g. start of time window, end of time window).

MI-15 {HIGH} The CDNI Metadata interface shall allow exchange of opaque metadata element, whose semantic is not defined in CDNI but established by private CDN agreement.

MI-16 {HIGH} The CDNI Metadata interface shall allow signaling of authorization checks and validation that are to be performed by the surrogate before delivery. For example, this could potentially include the need to validate information (e.g. Expiry time, Client IP address) required for access authorization.

MI-17 {MED} The CDNI Metadata interface should allow signaling of CDNI-relevant surrogate cache behavior parameters. For example, this could potentially include:

* control of whether the query string of HTTP URI is to be ignored by surrogate cache

* enforcement of caching directives by Downstream CDN that are different than the ones signalled in the HTTP headers (e.g. "Expires" field)

* rate-pacing by Downstream CDN for content delivery (e.g. Progressive Download)

MI-18 {HIGH} The CDNI Metadata interface shall provide indication of related content (e.g. HTTP Adaptive Bit Rate chunks) by the Content Collection ID (CCID) metadata. This could be used by the Downstream CDN for operations on the group of content. For example, this could potentially include:

* content acquisition for the entire set of files when one piece of content is requested

* local file management and storage bundles all the files for the content

* purging the entire set of files associated with the content

* logging of the delivery of the content for the session when at least one file in the set was delivered

MI-19 {MED} The CDNI Metadata interface should support an optional mechanism allowing the Upstream CDN to indicate to the Downstream CDN which CDNI Log fields are to be provided for all content items, for specific sets of content items, or for specific content items delivered using HTTP. A CDNI implementation that does not support this optional CDNI Metadata Distribution interface mechanism shall ignore this log format indication and generate CDNI logging format for HTTP Adaptive Streaming using the default set of CDNI Logging fields. (Note: This function may be part of the CDNI Metadata interface or the CDNI Control interface.)

MI-20 {MED} The CDNI Metadata interface should allow the Upstream CDN to signal to the Downstream CDN the Content Collection ID value for all, for specific sets of, or for specific content items delivered using HTTP. Whenever the Downstream CDN is instructed by the Upstream CDN to report the Content Collection ID field in the log records, the Downstream CDN is to use the value provided through the CDNI Metadata interface for the corresponding content. Note the Session ID field along with Content Collection ID may be used for HTTP Adaptive Streaming content.

MI-21 {MED} The CDNI Metadata interface should allow the Upstream CDN to signal to the Downstream CDN the Authorization Group ID value for all the related HTTP Adaptive Streaming content (i.e. manifest file and chunks). The authorization result of

a content (e.g. manifest file) is transferred over to related content (e.g. chunks).

MI-22 {HIGH} The CDNI Metadata interface shall support extensible format for CDNI metadata delivery from the Upstream CDN to the Downstream CDN.

8. CDNI Logging Interface Requirements

This section identifies the requirements related to the CDNI Logging interface (LI). We observe that while the CDNI Logging interface is currently discussed as a single "protocol", further analysis will determine whether the corresponding requirements are to be realized over a single interface and protocol, or over multiple interfaces and protocols.

LI-1 {HIGH} The CDNI logging architecture and interface shall ensure reliable transfer of CDNI logging information across CDNs.

LI-2 {HIGH} The CDNI Logging interface shall provide logging of deliveries and incomplete deliveries to User Agents performed by the Downstream CDN as a result of request redirection by the Upstream CDN.

LI-3 {MED} In the case of cascaded CDNs, the CDNI Logging interface should allow the Downstream CDN to report to the Upstream CDN logging for deliveries and incomplete deliveries performed by the Downstream CDN itself as well as logging for deliveries and incomplete deliveries performed by cascaded CDNs on behalf of the Downstream CDN.

LI-4 {HIGH} The CDNI Logging interface shall support batch/offline exchange of logging records.

LI-5 {MED} The CDNI Logging interface should also support an additional mechanism taking into account the timing constraints for some types of logging records (e.g. near-real time for monitoring and analytics applications).

LI-6 {HIGH} The CDNI Logging interface shall define a log file format and a set of fields to be exported for various CDNI logging events.

LI-7 {HIGH} The CDNI Logging interface shall define a transport mechanism to exchange CDNI Logging files.

LI-8 {MED} The CDNI Logging interface should allow a CDN to query another CDN for relevant current logging records (e.g. For on-demand access to real-time logging information).

LI-9 {LOW} The CDNI Logging interface may support aggregate/ summarized logs (e.g. total bytes delivered for a content regardless of individual User Agents to which it was delivered).

LI-10 {LOW} The CDNI Logging interface may support logging of performance data for deliveries to User Agents performed by the Downstream CDN as a result of request redirection by the Upstream CDN. Performance data may include various traffic statistics (the specific parameters are to be determined). The CDNI Logging interface may support the Upstream CDN to indicate the nature and contents of the performance data to be reported by the Downstream CDN.

LI-11 {MED} The CDNI Logging interface should support logging of consumed resources (e.g. storage, bandwidth) to the Upstream CDN for deliveries where content is stored by the Downstream CDN for delivery to User Agents. The information logged may include the type of storage (e.g., Origin, Intermediate, Edge, Cache) as well as the amount of storage (e.g., total GB, GB used, per time period, per content domain) all of which may impact the cost of the services.

LI-12 {MED} In the case of cascaded CDNs, the CDNI Logging interface should support the Downstream CDN to report consumed resources (e.g. storage, bandwidth) to the Upstream CDN where content is stored by the Downstream CDN itself as well as logging for storage resources when content storage is performed by cascaded CDNs on behalf of the Downstream CDN.

LI-13 {HIGH} The CDNI Logging interface shall support logging of deleted objects from the Downstream CDN to the Upstream CDN as a result of explicit delete requests on via the CDNI Control interface from the Upstream CDN.

LI-14 {HIGH} The CDNI Logging interface shall support the exchange of extensible log file formats to support proprietary information fields. These information fields shall be agreed upon ahead of time between the corresponding CDNs.

LI-15 {HIGH} The CDNI Logging interface shall allow a CDN to notify another CDN about which CDNI logging information is available for transfer and/or no longer available (e.g. it exceeded some logging retention period or some logging retention volume).

LI-16 {MED} The CDNI Logging interface should support the ability for the Downstream CDN to include the Content Collection ID and Session ID fields in CDNI log entries generated for HTTP Adaptive Streaming content.

LI-17 {MED} The CDNI Logging interface should provide privacy protection by not disclosing information that can be used to identify the user (e.g. method that anonymizes the IP address carried in the logging field). The use of the privacy protection mechanism is optional.

9. CDNI Security Requirements

This section identifies the requirements related to the CDNI security. Some of these are expected to affect multiple or all protocols.

SEC-1 {HIGH} All the CDNI interface shall support secure operation over unsecured IP connectivity (e.g. The Internet). This includes authentication, confidentiality, integrity protection as well as protection against spoofing and replay.

SEC-2 {HIGH} The CDNI solution shall provide sufficient protection against Denial of Service attacks. This includes protection against spoofed delivery requests sent by User Agents directly to a Downstream CDN attempting to appear as if they had been redirected by a given Upstream CDN when they have not.

SEC-3 {MED} The CDNI solution should be able to ensure that for any given request redirected to a Downstream CDN, the Downstream CDN can determine the Upstream CDN that redirected the request directly to the Downstream CDN (leading to that request being served by that CDN, or being further redirected).

SEC-4 {MED} The CDNI solution should be able to ensure that for any given transaction log generated by the Downstream CDN and communicated to an Upstream CDN, the Upstream CDN can confirm the transmitted log record corresponds to a request redirection by the Upstream CDN.

SEC-5 {LOW} The CDNI solution may provide a mechanism allowing an Upstream CDN that has credentials to acquire content from the CSP origin server (or another CDN), to allow establishment of credentials authorizing the Downstream CDN to acquire the content from the CSP origin server (or the other CDN) (e.g. In case the content cannot be acquired from the Upstream CDN).

10. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

11. Security Considerations

This document discusses CDNI security requirements in Section 9.

12. Contributors

This document reflects the contributions from the following authors:

Francois Le Faucheur

Cisco Systems

flefauch@cisco.com

Mahesh Viveganandhan

Cisco Systems

mvittal@cisco.com

Grant Watson

Alcatel-Lucent (Velocix)

gwatson@velocix.com

13. Acknowledgements

This document leverages the earlier work of the IETF CDI working group in particular as documented in [I-D.cain-request-routing-req],

[I-D.amini-cdi-distribution-reqs] and [I-D.gilletti-cdnp-aaa-reqs].

The authors would like to thank Gilles Bertrand, Christophe Caillet, Bruce Davie, Phil Eardley, Ben Niven-Jenkins, Agustin Schapira, Emile Stephan, Eric Burger, Susan He, Kevin Ma, Daryl Malas, Iuniana Oprescu, and Spencer Dawkins for their input. Serge Manning along with Robert Streijl, Vishwa Prasad, Percy Tarapore, Mike Geller, and Ramki Krishnan contributed to this document by addressing the requirements of the ATIS Cloud Services Forum.

Ray Brandenburg, Matt Caufield, and Gilles Bertrand provided valuable inputs for HTTP Adaptive Streaming, CDNI Metadata interface, and CDNI Logging interface, respectively.

Stephen Farrell, Adrian Farrel, Benoit Claise, Sean Turner, Christer Holmberg, and Carlos Pignataro provided review comments that helped improve the document.

14. References

14.1. Normative References

[I-D.ietf-cdni-framework]
Peterson, L. and B. Davie, "Framework for CDN Interconnection", draft-ietf-cdni-framework-07 (work in progress), November 2013.

[RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, September 2012.

14.2. Informative References

[ATIS-0800042]
"ATIS IPTV Content on Demand Service, https://www.atis.org/docstore/product.aspx?id=25670", December 2010.

[I-D.amini-cdi-distribution-reqs]
Amini, L., "Distribution Requirements for Content Internetworking", draft-amini-cdi-distribution-reqs-02 (work in progress), November 2001.

[I-D.cain-request-routing-req]
Cain, B., "Request Routing Requirements for Content Internetworking", draft-cain-request-routing-req-03 (work in progress), November 2001.

[I-D.gilletti-cdnp-aaa-reqs]
"CDI AAA Requirements,
draft-gilletti-cdnp-aaa-reqs-01.txt", June 2001.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[RFC6770] Bertrand, G., Stephan, E., Burbridge, T., Eardley, P., Ma, K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", RFC 6770, November 2012.

[RTMP] "Adobe's Real Time Messaging Protocol, http://www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf", December 2012.

Authors' Addresses

Kent Leung (editor)
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134
U.S.A.

Phone: +1 408 526 5030
Email: kleung@cisco.com

Yiu Lee (editor)
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: yiu_lee@cable.comcast.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: March 25, 2012

G. Bertrand, Ed.
E. Stephan
France Telecom - Orange
G. Watson
T. Burbridge
P. Eardley
BT
K. Ma
Azuki Systems

September 22, 2011

Use Cases for Content Delivery Network Interconnection
draft-ietf-cdni-use-cases-00

Abstract

Content Delivery Networks (CDNs) are commonly used for improving the footprint and the end-user experience of a content delivery service, at a reasonable cost. This document outlines real world use-cases (not technical solutions) for interconnecting CDNs. It can be used to provide guidance to the CDNI WG about the interconnection arrangements to be supported and to validate the requirements of the various CDNI interfaces.

This document describes a number of use cases that motivate CDN Interconnection. It represents a work in progress and may be extended later to cover additional use-cases.

Table of Contents

1. Introduction

Content Delivery Networks (CDNs) are commonly used for improving the footprint and the end-user experience of a content delivery service, at a reasonable cost. This document outlines real world use-cases (not technical solutions) for interconnecting CDNs. It can be used to provide guidance to the CDNI WG about the interconnection arrangements to be supported and to validate the requirements of the various CDNI interfaces.

This document describes a number of use cases that motivate CDN Interconnection. It represents a work in progress and may be extended later to cover additional use-cases.

This document identifies the main motivations for a CDN Provider to interconnect its CDN:

- CDN Footprint Extension Use Cases (Section 2)
- CDN Offload Use Cases (Section 3)
- CDN Capability Use Cases (Section 4)

Then, the document highlights the need for interoperability to exchange and enforce content delivery policies (Section 5).

1.1. Terminology

We adopt the terminology described in [I-D.ietf-cdni-problem-statement], [RFC3466], and [RFC3568], except for the terms defined below.

Authoritative CDN (aCDN):

A CDN provider contracted by the CSP for delivery of content by its CDN or by its downstream CDNs.

Access CDN:

A CDN that is connected to the end-user’s access and has information about the end-user’s profile and access capabilities.

Delivering CDN:

The CDN that delivers the requested content asset to the end-user. In particular, the delivering CDN can be an access CDN.

[Ed. Note: Definition of recursive and iterative request routing

should be added to [I-D.davie-cdni-framework].]

CDN Interconnection:

Relationship between two CDNs that enables a CDN to provide content delivery services on behalf of another CDN. It relies on a set of interfaces over which two CDNs communicate in order to achieve the delivery of content to end-users by one CDN (the downstream CDN) on behalf of another CDN (the upstream CDN).

CDN peering: A business relation between two CDN providers based on one or more CDN Interconnections.

1.2. Abbreviations

[Ed. Note: List of abbreviations to be updated later]

- CSP: Content Service Provider
- dCDN: downstream CDN
- NSP: Network Service Provider
- QoE: Quality of Experience
- QoS: Quality of Service
- SLA: Service Level Agreement
- uCDN: upstream CDN
- UA: User Agent
- UE: User Equipment
- VoD: Video on Demand
- WiFi: Wireless Fidelity

1.3. Rationale for Multi-CDN Systems

Content Delivery Networks (CDNs) are used to deliver content because they can:

- improve the experience for the End User; for instance delivery has lower latency and better robustness,

- o reduce the operator's costs; for instance lower delivery cost (reduced bandwidth usage) for cacheable content,

- o reduce the Content Service Provider costs, such as datacenter capacity, space, and electricity consumption.

Indeed, many network service providers and enterprise service providers are deploying or have deployed their own CDNs. Despite the potential benefits of interconnecting CDNs, today each CDN is a standalone network. The objective of CDN interconnection is to overcome this restriction: the interconnected CDNs should be able to collectively behave as a single delivery infrastructure.

Let's take an example, as depicted in Figure 1. Two CDN Providers establish a CDN Interconnection. The Content Service Provider CSP-1 reaches an agreement with CDN Provider A for the delivery of its content. CDN Provider A and CDN Provider B agree to interconnect their CDNs. When a User Agent that is connected to CDN Provider B's network requests Content from CSP-1, the Content is actually delivered from CDN-B, because handling of requests for CSP-1's Content has been delegated as part of the CDN Interconnection agreement. The End User benefits through a better quality of experience, because the Content is delivered from a nearby Surrogate. CDN Provider A benefits because it doesn't need to deploy such an extensive CDN, whilst CDN Provider B receives some compensation for the delivery. CSP-1 benefits because it only needs to make one business agreement and one physical connection, with CDN Provider A, but its End Users get a service quality as though CSP-1 had also gone to the trouble of making a business agreement with CDN Provider B.

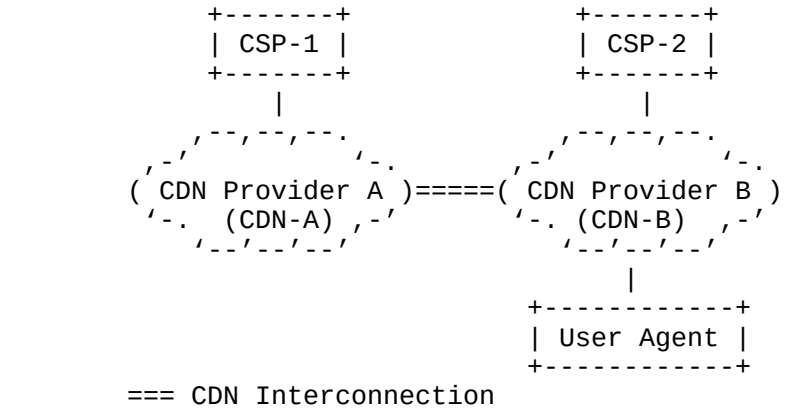


Figure 1

To extend the example, another Content Service Provider, CSP-3, may also reach an agreement with CDN Provider A. But it does not want its Content to be distributed by CDN Provider B, for example, CSP-3 may not have distribution rights in the country where CDN Provider B operates. This example illustrates that policy considerations are an important part of CDNI.

1.4. The Need for CDNI Standards

Existing CDN interfaces are proprietary and have often been designed for intra-CDN/intra-domain operations. So an external CDN typically cannot use them, especially if the two CDNs rely on different implementations. Nevertheless, [I-D.bertrand-cdni-experiments] shows that some level of CDN interconnection can be achieved experimentally without standardized interfaces between the CDNs. However, the methods used in these experiments are hardly usable in an operational context, because they suffer from several limitations in terms of functionalities, scalability, and security level.

The aim of IETF CDNI WG's solution is therefore to overcome such shortcomings; a full list of requirements is being developed in [I-D.ietf-cdni-requirements].

2. Footprint Extension Use Cases

Footprint extension is expected to be a major use case for CDN interconnection.

2.1. Geographic Extension

In this use case, the CDN Provider wants to extend the geographic distribution that it can offer to its CSPs:

- without compromising the quality of delivery,
- without incurring additional transit and other network costs that would result from serving content from geographically or topologically remote surrogates.

If there are several CDN Providers that have a geographically limited footprint (e.g., restricted to one country), or do not serve all end-users in a geographic area, then interconnecting their CDNs enables CDN Providers to provide their services beyond their own footprint.

As an example, suppose a French CSP wants to distribute its TV programs to End Users located in France and various countries in North Africa. It asks a French CDN Provider to deliver the content.

The French CDN Provider's network only covers France, so it makes an agreement with another CDN Provider that covers North Africa. Overall, from the CSP's perspective the French CDN Provider provides a CDN service for both France and North Africa.

In addition to video, this use case applies to other types of content such as automatic software updates (browser updates, operating system patches, virus database update, etc).

2.2. Inter-Affiliates Interconnection

In the previous section, we have described the case of geographic extension between CDNs operated by different entities. A large CDN Provider may also operate CDNs from several subsidiaries (which may rely on different CDN solutions, see Section 4.2). In certain circumstances, the CDN Provider needs to make its CDNs interoperate to provide a consistent service to its customers on its whole footprint.

2.3. Nomadic Users

In this scenario a CSP wishes to allow users who move to other geographic regions to continue to access their content. The motivation in this case is to allow nomadic users to maintain access with consistent quality of experience, rather than to allow all residents within a region access to the content.

[Ed. Note: expand on which CDNs need to be interconnected to address the use case (ie with CDN of "home NSP" interconnected to CDN of visited NSP). Add a picture for clarifying the text. Use the term "TV everywhere"?]

This use case covers situations like users moving between different CDN Providers within the same geographic region, or users switching between different devices, as discussed in Section 4.

3. Offload Use Cases

3.1. Overload Handling and Dimensioning

A CDN is likely to be dimensioned to support the prime-time traffic. However, unexpected spikes in content popularity may drive load beyond the expected peak. The prime recurrent time peaks of content distribution may differ between two CDNs. Taking advantage of the different traffic peak times, a CDN may interconnect with another CDN to increase its effective capacity during the peak of traffic. This brings dimensioning savings to the CDNs as they can use the resources

of each other during their respective peaks of activity..

Offload also applies to planned situations where a CDN Provider needs CDN capacities in a particular region during a short period of time. For example, a CDN can offload traffic to another CDN during a specific maintenance operation or for covering the distribution of a special event. For instance, consider a TV-channel which has exclusive distribution rights on a major event, such as a celebrities' wedding, or a major sport competitions. The CDNs that the TV-channel uses for delivering the content related to this event are likely to experience a flash crowd during the event and to need offloading traffic, while other CDNs will support a more usual traffic load and be able to handle the offloaded traffic load.

3.2. Resiliency

3.2.1. Failure of Content Delivery Resources

It is important for CDNs to be able to guarantee service continuity during partial failures (e.g., failure of some Surrogates). In partial failure scenarios, a CDN Provider could redirect some requests towards another CDN, which must be able to serve the redirected requests or, depending on traffic management policies, to forward these requests to the CSP's origin server.

3.2.2. Failure of Content Acquisition

Source content acquisition is typically handled in one of two ways:

- CDN origin, where a downstream CDN acquires content from an upstream CDN, and the authoritative CDN acquires content from an origin server of the CSP, or

- Other origin, where the CDNs acquire content directly from an origin server outside the uCDN.

Resiliency may be required against failure to ingest content. If a CDN is unable to retrieve the content, it may be that the CSP's origin server is inaccessible to only this CDN, in which case redirection of the end-users to an alternative CDN may circumvent the problem. A CSP may also choose to specify one or more backup origin servers.

4. CDN Capability Use Cases

4.1. Device and Network Technology Extension

In this use case, the CDN Provider may have the right geographic footprint, but may wish to extend the supported range of devices and User Agents or the range of supported delivery technologies. In this case, a CDN Provider may interconnect with a CDN that offers services:

- that its own CDN is not able to support or,
- that the CDN provider is not willing to provide.

The following examples illustrate this use case:

1. CDN-A cannot support a specific delivery protocol. For instance, CDN-A may interconnect with CDN-B to serve a proportion of its traffic that requires HTTPS. CDN-A may use CDN-B's footprint (which may overlap with its own) to deliver HTTPS without needing to deploy its own infrastructure. This case could also be true of other formats, delivery protocols (RTMP, RTSP, etc.) and features (specific forms of tokenization, per session encryption, etc.).

2. CDN-A has footprint covering traditional fixed line broadband and wants to extend coverage to mobile devices. In this case, CDN-A may contract and interconnect with CDN-B who has both:

   * physical footprint inside the mobile network,

   * the ability to deliver content over a protocol that is required by specific mobile devices and not supported by CDN-A.

   In this case also it may be that CDN-B provides other features related to adapting the content.

These cases can apply to many CDN features that a given CDN provider may not be able to support or not be willing to invest in, and thus, that the CDN provider would delegate to another CDN.

4.2. Technology and Vendor Interoperability

A CDN Provider may deploy a new CDN to run alongside its existing CDN, as a simple way of migrating its CDN service to a new technology. A CDN Provider may have a multi-vendor strategy for its CDN deployment. A CDN Provider may want to deploy a separate CDN for a particular CSP or a specific network. In all these circumstances, CDNI benefits the CDN Provider, as it simplifies or automates some

inter-CDN operations (e.g., migrating the request routing function progressively).

4.3. QoE and QoS Improvement

Some CSPs are willing to pay a premium for enhanced delivery of Content to their End Users. In some cases, even if the CDN Provider could deliver the content to the end users, it cannot meet the CSP's service level requirements. So, it makes a CDN Interconnection agreement with another CDN Provider that can provide the expected quality of experience to the end-user, for instance an Access CDN able to deliver content from Surrogates located closer to the end-user.

5. Policy Enforcement

For the interconnection use cases described in previous sections, the delegation of content delivery may be dependent upon the ability to delegate delivery policy enforcement as well. CSPs may rely on the ability to place delivery restriction on sets of content, which are provided by existing CDNs. While the ability to support these features across interconnected CDNs is desirable, that may not always be feasible. It is important to be able to detect or define when these features cannot be enforced.

5.1. Content Availability

The content distribution policies that a CSP attaches to a content asset depend on many criteria. For instance, distribution policies for audiovisual content often combine:

- temporal constraints (e.g., available for 24 hours, available 28 days after DVD release, etc.),
- resolution-based constraints (e.g., high definition vs. standard definition), and
- geolocation-based constraints (e.g., per country).

5.1.1. Geo-location Restrictions

"Geo-blocking" rules may specify:

- the geographic regions where content can be delivered from (i.e. the location of the Surrogates), or

- o geographic locations where content can be delivered to (i.e., the location of the End Users).

If a default value of "geo-blocking rules not supported" is set, the CSP may wish to deny all access to the content, or blacklist specific dCDNs which lack support for these features.

5.1.2. Temporal Restrictions

Time-based rules may specify:

- o an activation time (i.e., the time when the content should become available for delivery),
- o a deactivation time (i.e., time after which the content should no longer be delivered), or
- o an expiration time (i.e., the time at which the content files should be expunged from all CDN storage).

If a default value of "time-based rules not supported" is set, the CSP may wish to deny all access to the content, or blacklist specific dCDNs which lack support for these features.

5.1.3. Content Encoding Restrictions

[Ed. Note: Section to be removed? reworded?]

Encoding-based rules may specify:

- o a subset of encodings deliverable to specific devices,
- o a subset of encodings deliverable though a specific NSP, or
- o a subset of encodings deliverable to users based on a subscription or quality of service levels.

[Ed. Note: FLF The first bullet only makes sense if the solution supports transcoding/transrating, which I don’t know if it will be supported in Initial scope. The last bullet does not make sense to me as we do not want the CDNs to be aware of any user subscription levels (ie only CSP is aware of user subscription level)."]

If a default value of "encoding-based rules not supported" is set, the CSP may wish to deny all access to the content, or blacklist specific dCDNs which lack support for these features.

5.2. Branding

There are situations where one CDN Provider cannot or does not want to operate all the functions of a uCDN, e.g., a CDN exchange, which handles request routing (and possibly log retrieval) but delegates all content delivery to the surrogates of other dCDNs. Preserving the branding of the CSP throughout delivery is often important to the CSP. CSPs may desire to offer content services under their own name, even when the associated CDN service involves other CDSPs. The CSP may request that the name of the CDSPs does not appear in the URLs and may wish to specify a specific brand related tag to appear in the URLs. Similarly, in offload situations, the uCDN might want to offer CDN services under its own branding.

If a default value of "branding rules not supported" is set, the CSP may wish to deny all access to the content, or blacklist specific dCDNs which lack support for these features.

5.3. Secure Access

Many protocols exist for delivering content to End Users. CSPs may often wish to dictate a specific protocol or set of protocols which are acceptable for delivery of their content, especially in the case where content protection or user authentication is required (e.g., must use HTTPS and not HTTP, or must use URL hashing, etc.).

If a default value of "secure access rules not supported" is set, the CSP may wish to deny all access to the content, or blacklist specific dCDNs which lack support for these features.

6. Open issues

The section about nomadic users must be clarified

The section about Content Encoding Restrictions requires a discussion: must the CDN enforce such restrictions?

7. Contributors

[Ed. Note: long list of co-authors. As per current practice, you would want to split that into "editors" and "contributors".]

The following people have strongly contributed to this specification’s content:

8. Acknowledgments

The authors would like to thank Kent Leung, Francois Le Faucheur and Ben Niven-Jenkins for lively discussions, as well as for their reviews and comments on the mailing list.

They also thank the contributors of the EU FP7 OCEAN and ETICS projects for valuable inputs.

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

CDN Interconnection, as described in this document, has a wide variety of security issues that should be considered. The security issues fall into three general categories:

- CSP Trust: where the CSP may have negotiated service level agreements for delivery quality of service with the uCDN, and/or configured distribution policies (e.g., geo-restrictions, availability windows, or other licensing restrictions), which it assumes will be upheld by dCDNs to which the uCDN delegates requests. Furthermore, billing and accounting information must be aggregated from dCDNs with which the CSP may have no direct business relationship. These situations where trust is delegated must be handled in a secure fashion to ensure CSP confidence in the CDN interconnection.

- Client Transparency: where the client device or application which connects to the CDN must be able to interact with any dCDN using its existing security and DRM protocols (e.g., cookies, certificate-based authentication, custom DRM protocols, URL signing algorithms, etc.) in a transparent fashion.

- CDN Infrastructure Protection: where the dCDNs must be able to identify and validate delegated requests, in order to prevent unauthorized use of the network and to be able to properly bill for delivered content. A dCDN may not wish to advertise that it has access to or is carrying content for the uCDN or CSP, especially if that information may be used to enhance denial of service attacks. In general, CDNI interfaces and protocols should minimize overhead for dCDNs.

This document focuses on the motivational use cases for CDN

Interconnection, and does not analyze these threats in detail.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

[I-D.bertrand-cdni-experiments]
Bertrand, G., Faucheur, F., and L. Peterson, "Content Distribution Network Interconnection (CDNI) Experiments", draft-bertrand-cdni-experiments-01 (work in progress), August 2011.

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-00 (work in progress), September 2011.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-00 (work in progress), September 2011.

[I-D.ma-cdni-publisher-use-cases]
Nair, R. and K. Ma, "Content Distribution Network Interconnection (CDNI) Publisher Use", draft-ma-cdni-publisher-use-cases-00 (work in progress), March 2011.

[I-D.watson-cdni-use-cases]
Watson, G., "CDN Interconnect Use Cases", draft-watson-cdni-use-cases-00 (work in progress), January 2011.

[RFC3466] Day, M., Cain, B., Tomlinson, G., and P. Rzewski, "A Model for Content Internetworking (CDI)", RFC 3466,

February 2003.

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

Authors' Addresses

Gilles Bertrand (editor)
France Telecom - Orange
38-40 rue du General Leclerc
Issy les Moulineaux, 92130
FR

Phone: +33 1 45 29 89 46
Email: gilles.bertrand@orange.com

Stephan Emile
France Telecom - Orange
2 avenue Pierre Marzin
Lannion F-22307
France

Email: emile.stephan@orange.com

Grant Watson
BT
pp GDC 1 PP14, Orion Building, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: grant.watson@bt.com

Trevor Burbridge
BT
B54 Room 70, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: trevor.burbridge@bt.com

Philip Eardley
BT
B54 Room 77, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: philip.eardley@bt.com

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

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

Internet Engineering Task Force
Internet-Draft
Obsoletes: 3570 (if approved)
Intended status: Informational
Expires: February 10, 2013

G. Bertrand, Ed.
E. Stephan
France Telecom - Orange
T. Burbridge
P. Eardley
BT
K. Ma
Azuki Systems, Inc.
G. Watson
Alcatel-Lucent (Velocix)

August 9, 2012

Use Cases for Content Delivery Network Interconnection
draft-ietf-cdni-use-cases-10

Abstract

Content Delivery Networks (CDNs) are commonly used for improving the End User experience of a content delivery service while keeping cost at a reasonable level. This document focuses on use cases that correspond to identified industry needs and that are expected to be realized once open interfaces and protocols supporting interconnection of CDNs are specified and implemented. The document can be used to motivate the definition of the requirements to be supported by CDN Interconnection (CDNI) interfaces. It obsoletes RFC 3570.

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 10, 2013.

Copyright Notice

Table of Contents

1. Introduction

Content Delivery Networks (CDNs) are commonly used for improving the End User experience of a content delivery service while keeping cost at a reasonable level. This document focuses on use cases that correspond to identified industry needs and that are expected to be realized once open interfaces and protocols supporting interconnection of CDNs are specified and implemented. The document can be used to motivate the definition of the requirements (as documented in [I-D.ietf-cdni-requirements]) to be supported by the set of CDN Interconnection (CDNI) interfaces defined in [I-D.ietf-cdni-problem-statement].

RFC 3570 described slightly different terminologies and models for "Content Internetworking (CDI)". The present document obsoletes RFC 3570 to avoid confusion.

This document identifies the main motivations for a CDN Provider to interconnect its CDN:

- CDN Footprint Extension Use Cases (Section 2)
- CDN Offload Use Cases (Section 3)
- CDN Capability Use Cases (Section 4)

Then, the document highlights the need for interoperability in order to exchange and enforce content delivery policies (Section 5).

1.1. Terminology

In this document, the first letter of each CDNI-specific term is capitalized. We adopt the terminology described in [I-D.ietf-cdni-problem-statement].

We extend this terminology with the following terms.

Access CDN:

A CDN that includes Surrogates in the same administrative network as the end-user. Such CDN can use accurate information on the End User’s network context to provide valued-added Content Delivery Services to Content Service Providers.

1.2. Abbreviations

- CDN: Content Delivery Network also known as Content Distribution Network
- CSP: Content Service Provider
- dCDN: downstream CDN
- DNS: Domain Name System
- EU: End User
- ISP: Internet Service Provider
- NSP: Network Service Provider
- QoE: Quality of Experience
- QoS: Quality of Service
- uCDN: upstream CDN
- URL: Uniform Resource Locator
- WiFi: Wireless local area network (WLAN) based on IEEE 802.11

1.3. Rationale for CDN Interconnection

Content Delivery Networks (CDNs) are used to deliver content because they can:

- improve the experience for the End User; for instance delivery has lower latency (decreased round-trip-time and higher throughput between the user and the delivery server) and better robustness (ability to use multiple delivery servers),
- reduce the network operator’s costs; for instance, lower delivery cost (reduced bandwidth usage) for cacheable content,
- reduce the Content Service Provider’s (CSP) internal infrastructure costs, such as datacenter capacity, space, and electricity consumption, as popular content is delivered externally through the CDN rather than through the CSP’s own servers.

Indeed, many Network Service Providers (NSPs) and enterprise service providers are deploying or have deployed their own CDNs. Despite the potential benefits of interconnecting CDNs, today each CDN is a standalone network. The objective of CDN Interconnection is to

overcome this restriction: the interconnected CDNs should be able to collectively behave as a single delivery infrastructure.

An example is depicted in Figure 1, where two CDN Providers establish a CDN Interconnection. The Content Service Provider CSP-1 reaches an agreement with CDN Provider 'A' for the delivery of its content. Independently, CDN Provider 'A' and CDN Provider 'B' agree to interconnect their CDNs.

When a given User Agent requests content from CSP-1, CDN-A may consider that delivery by CDN-B is appropriate, for instance, because CDN-B is an Access CDN and the user is directly attached to it. Through the CDN Interconnection arrangements put in place between CDN-A and CDN-B (as a result of the CDN Interconnection agreement established between CDN Provider 'A' and CDN Provider 'B'), CDN-A can redirect the request to CDN-B and the content is actually delivered to the User Agent by CDN-B.

The End User benefits from this arrangement through a better Quality of Experience (QoE, see [RFC6390]), because the content is delivered from a nearby Surrogate (e.g., lower latency, bottlenecks avoided). CDN Provider 'A' benefits because it does not need to deploy such an extensive CDN, whilst CDN Provider 'B' may receive some compensation for the delivery. CSP-1 benefits because it only needs to make one business agreement and one technical arrangement with CDN Provider 'A', but its End Users get a service quality as though CSP-1 had also gone to the trouble of making a business agreement and technical arrangement with CDN Provider 'B'.

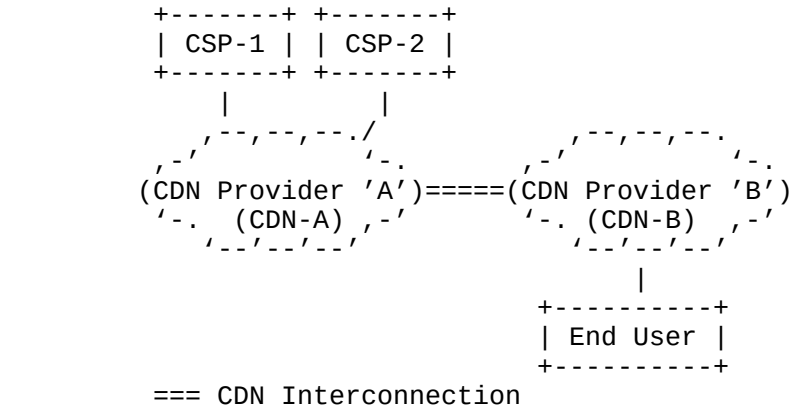


Figure 1

To extend the example, another Content Service Provider, CSP-2, may also reach an agreement with CDN Provider 'A'. However, CSP-2 may not want its content to be distributed by CDN Provider B; for example, CSP-2 may not want to distribute its content in the area where CDN Provider 'B' operates. This example illustrates that policy considerations are an important part of CDNI.

2. Footprint Extension Use Cases

Footprint extension is expected to be a major use case for CDN Interconnection.

2.1. Geographic Extension

In this use case, the CDN Provider wants to extend the geographic distribution that it can offer to its CSPs:

- without compromising the quality of delivery,
- without incurring additional transit and other network costs that would result from serving content from geographically or topologically remote Surrogates,
- without incurring the cost of deploying and operating Surrogates and the associated CDN infrastructure that may not be justified in the corresponding geographic region (e.g., because of relatively low delivery volume, or conversely because of the high investments that would be needed to satisfy the high volume).

If there are several CDN Providers that have a geographically limited footprint (e.g., restricted to one country), or do not serve all End Users in a geographic area, then interconnecting their CDNs enables these CDN Providers to provide their services beyond their own footprint.

As an example, suppose a French CSP wants to distribute its TV programs to End Users located in France and various countries in North Africa. It asks a French CDN Provider to deliver the content. The French CDN Provider's network only covers France, so it makes an agreement with another CDN Provider that covers North Africa. Overall, from the CSP's perspective the French CDN Provider provides a CDN service for both France and North Africa.

In addition to video, this use case applies to other types of content such as automatic software updates (browser updates, operating system patches, virus database update, etc).

2.2. Inter-Affiliates Interconnection

The previous section describes the case of geographic extension between CDNs operated by different entities. A large CDN Provider may have several subsidiaries that also each operate their own CDN (which may rely on different CDN technologies, see Section 4.2). In certain circumstances, the CDN Provider needs to make these CDNs interoperate to provide a consistent service to its customers on the whole collective footprint.

2.3. ISP Handling of Third-Party Content

Consider an ISP carrying to its subscribers a lot of content that comes from a third party CSP and that is injected into the ISP's network by an Authoritative CDN Provider. There are mutual benefits to the ISP (acting as an Access CDN), the Authoritative CDN, and the CSP that would make a case for establishing a CDNI agreement. For example:

- Allow the CSP to offer improved QoE and QoE services to subscribers, for example, reduced content startup time or increased video quality and resolution of adaptive streaming content.

- Allow the Authoritative CDN to reduce hardware capacity and footprint, by using the ISP caching and delivery capacity.

- Allow the ISP to reduce traffic load on some segments of the network by caching inside of the ISP network.

- Allow the ISP to influence and/or control the traffic ingress points.

- Allow the ISP to derive some incremental revenue for transport of the traffic and to monetize QoE services.

2.4. Nomadic Users

In this scenario, a CSP wishes to allow End Users who move between access networks to continue to access their content. The motivation of this case is to allow nomadic End Users to maintain access to content with a consistent QoE, across a range of devices and/or geographic regions.

This use case covers situations like:

- End Users moving between different access networks, which may be located within the same geographic region or different geographic

regions,

- End Users switching between different devices or delivery technologies, as discussed in Section 4.

Consider the following example, illustrated in Figure 2: End User A has subscription to a broadband service from ISP A, her "home ISP". ISP A hosts CDN-A. Ordinarily, when End User A accesses content via ISP A (her "home ISP") the content is delivered from CDN-A, which in this example is within ISP A's network.

However, while End User A is not connected to ISP A's network, for example, because it is connected to a WiFi provider or mobile network, End User A can also access the same content. In this case, End User A may benefit from accessing the same content but delivered by an alternate CDN (CDN-B), in this case, hosted in the network of the WiFi or mobile provider (ISP B), rather than from CDN-A in ISP A's network.

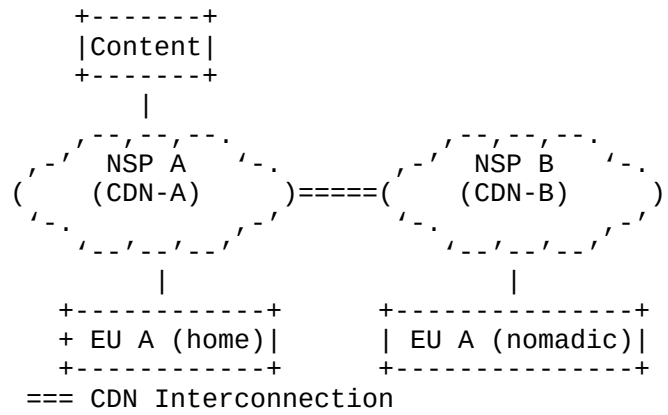


Figure 2

Though the content of CSP A is not accessible by typical End Users of CDN-B, End User A is able to gain access to their "home" content (i.e., the content of CSP A) through the alternate CDN (CDN-B).

Depending on CSP's content delivery policies (see Appendix A.1), a user moving to a different geographic region may be subject to geo-blocking content delivery restrictions. In this case, he/she may not be allowed to access some pieces of content.

3. Offload Use Cases

3.1. Overload Handling and Dimensioning

A CDN is likely to be dimensioned to support an expected maximum traffic load. However, unexpected spikes in content popularity (flash crowd) may drive load beyond the expected peak. The prime recurrent time peaks of content distribution may differ between two CDNs. Taking advantage of the different traffic peak times, a CDN may interconnect with another CDN to increase its effective capacity during the peak of traffic. This brings dimensioning savings to the CDNs as they can use the resources of each other during their respective peaks of activity.

Offload also applies to planned situations where a CDN Provider needs CDN capacity in a particular region during a short period of time. For example, a CDN can offload traffic to another CDN during a specific maintenance operation or for covering the distribution of a special event, as in the scenario depicted in Figure 3. For instance, consider a TV-channel which is the distributor for a major event, such as a celebrities' wedding, or a major sport competition and this TV-channel has contracted particular CDNs for the delivery. The CDNs (CDN-A and CDN-B) that the TV-channel uses for delivering the content related to this event are likely to experience a flash crowd during the event and to need offloading traffic, while other CDNs (CDN-C) will support a more usual traffic load and be able to handle the offloaded traffic.

In this use case, the Delivering CDN on which requests are offloaded should be able to handle the offloaded requests. Therefore, the uCDN might require information on the dCDNs to be aware of the amount of traffic it can offload to every dCDN.

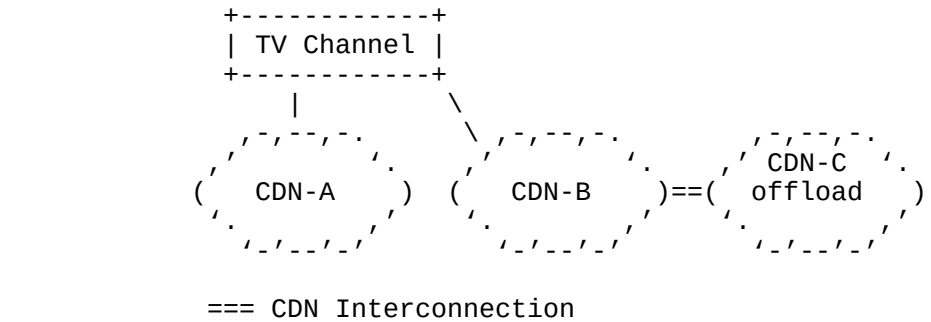


Figure 3

3.2. Resiliency

3.2.1. Failure of Content Delivery Resources

It is important for CDNs to be able to guarantee service continuity during partial failures (e.g., failure of some Surrogates). In partial failure scenarios, a CDN Provider has at least three options:

1. if possible, use internal mechanisms to redirect traffic on surviving equipment,

2. depending on traffic management policies, forward some requests to the CSP's origin servers, and

3. redirect some requests toward another CDN, which must be able to serve the redirected requests.

The last option is a use case for CDNI.

3.2.2. Content Acquisition Resiliency

Source content acquisition may be handled in one of two ways:

- CSP origin, where a CDN acquires content directly from the CSP's origin server, or

- CDN origin, where a downstream CDN acquires content from a Surrogate within an upstream CDN.

The ability to support content acquisition resiliency, is an important use case for interconnected CDNs. When the content acquisition source fails, the CDN might switch to another content acquisition source. Similarly, when several content acquisition sources are available, a CDN might balance the load between these multiple sources.

Though other server and/or DNS load balancing techniques may be employed in the network, interconnected CDNs may have a better understanding of origin server availability and be better equipped to both distribute load between origin servers and attempt content acquisition from alternate content sources when acquisition failures occur. When normal content acquisition fails, a CDN may need to try other content source options, e.g.:

- an upstream CDN may acquire content from an alternate CSP origin server,

- a downstream CDN may acquire content from an alternate Surrogate within an upstream CDN,
- a downstream CDN may acquire content from an alternate upstream CDN, or
- a downstream CDN may acquire content directly from the CSP's origin server.

Though content acquisition protocols are beyond the scope of CDNI, the selection of content acquisition sources should be considered and facilitated.

4. CDN Capability Use Cases

4.1. Device and Network Technology Extension

In this use case, the CDN Provider may have the right geographic footprint, but may wish to extend the supported range of devices and User Agents or the supported range of delivery technologies. In this case, a CDN Provider may interconnect with a CDN that offers services:

- that the CDN Provider is not willing to provide or,
- that its own CDN is not able to support.

The following examples illustrate this use case:

1. CDN-A cannot support a specific delivery protocol. For instance, CDN-A may interconnect with CDN-B to serve a proportion of its traffic that requires HTTPS [RFC2818]. CDN-A may use CDN-B's footprint (which may overlap with its own) to deliver HTTPS without needing to deploy its own infrastructure. This case could also be true of other formats, delivery protocols (RTMP, RTSP, etc.) and features (specific forms of authorization such as tokens, per session encryption, etc.).

2. CDN-A has footprint covering traditional fixed line broadband and wants to extend coverage to mobile devices. In this case, CDN-A may contract and interconnect with CDN-B who has both:

    * physical footprint inside the mobile network,
    * the ability to deliver content over a protocol that is required by specific mobile devices.

3. CDN-A only supports IPv4 within its infrastructure but wants to deliver content over IPv6. CDN-B supports both IPv4 and IPv6 within its infrastructure. CDN-A interconnects with CDN-B to serve out its content over native IPv6 connections.

These cases can apply to many CDN features that a given CDN Provider may not be able to support or not be willing to invest in, and thus, that the CDN Provider would delegate to another CDN.

4.2. Technology and Vendor Interoperability

A CDN Provider may deploy a new CDN to run alongside its existing CDN, as a simple way of migrating its CDN service to a new technology. In addition, a CDN Provider may have a multi-vendor strategy for its CDN deployment. Finally, a CDN Provider may want to deploy a separate CDN for a particular CSP or a specific network. In all these circumstances, CDNI benefits the CDN Provider, as it simplifies or automates some inter-CDN operations (e.g., migrating the request routing function progressively).

4.3. QoE and QoS Improvement

Some CSPs are willing to pay a premium for enhanced delivery of content to their End Users. In some cases, even if the CDN Provider could deliver the content to the End Users, it cannot meet the CSP's service level requirements. As a result, the CDN Provider may establish a CDN Interconnection agreement with another CDN Provider that can provide the expected QoE to the End User, e.g., via an Access CDN able to deliver content from Surrogates located closer to the End User and with the required service level.

5. Enforcement of Content Delivery Policy

An important aspect common to all the above use cases is that CSPs typically want to enforce content delivery policies. A CSP may want to define content delivery policies that specify when, how, and/or to whom the CDN delivers content. These policies apply to all interconnected CDNs (uCDNs and dCDNs) in the same or similar way that a CSP can define content delivery policies for content delivered by a single, non-interconnected CDN. Appendix A provides examples of CSP defined policies.

6. Acknowledgments

The authors would like to thank Kent Leung, Francois Le Faucheur, Ben Niven-Jenkins, and Scott Wainner for lively discussions, as well as

for their reviews and comments on the mailing list.

They also thank the contributors of the EU FP7 OCEAN and ETICS projects for valuable inputs.

Finally, the authors acknowledge the work of the former CDI working group, which is now obsoleted to avoid confusion.

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

This document focuses on the motivational use cases for CDN Interconnection, and does not analyze the associated threats. Those are discussed in [I-D.ietf-cdni-problem-statement]. Appendix A.2 provides example security policies that CSPs might impose on CDNs to mitigate the threats.

9. References

9.1. Normative References

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-08 (work in progress), June 2012.

9.2. Informative References

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-03 (work in progress), June 2012.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, October 2011.

Appendix A. Content Service Providers' Delivery Policies

CSPs commonly apply different delivery policies to given sets of content assets delivered through CDNs. Interconnected CDNs need to support these policies. This annex presents examples of CSPs' delivery policies and their consequences on CDNI operations.

A.1. Content Delivery Policy Enforcement

The content distribution policies that a CSP attaches to a content asset may depend on many criteria. For instance, distribution policies for audiovisual content often combine constraints of varying levels of complexity and sophistication, e.g.:

- temporal constraints (e.g., available for 24 hours, available 28 days after DVD release, etc.),
- user agent platform constraints (e.g., mobile device platforms, desktop computer platforms, set-top-box platforms, etc.),
- resolution-based constraints (e.g., high definition vs. standard definition encodings),
- user agent identification or authorization,
- access network constraints (e.g., per NSP), and
- IP geo-blocking constraints (e.g., for a given coverage area).

CSPs may use sophisticated policies in accordance to their business model. However, the enforcement of those policies does not necessarily require that the delivery network understand the policy rationales or how policies apply to specific content assets. Content delivery policies may indeed be distilled into simple rules which can be commonly enforced across all dCDNs. These rules may influence dCDN delegation and Surrogate selection decisions, for instance, to ensure that the specific rules (e.g. time-window, geo-blocking, pre-authorization validation) can indeed be enforced by the delivering CDN. In turn, this can guarantee to the CSP that content delivery policies are properly applied.

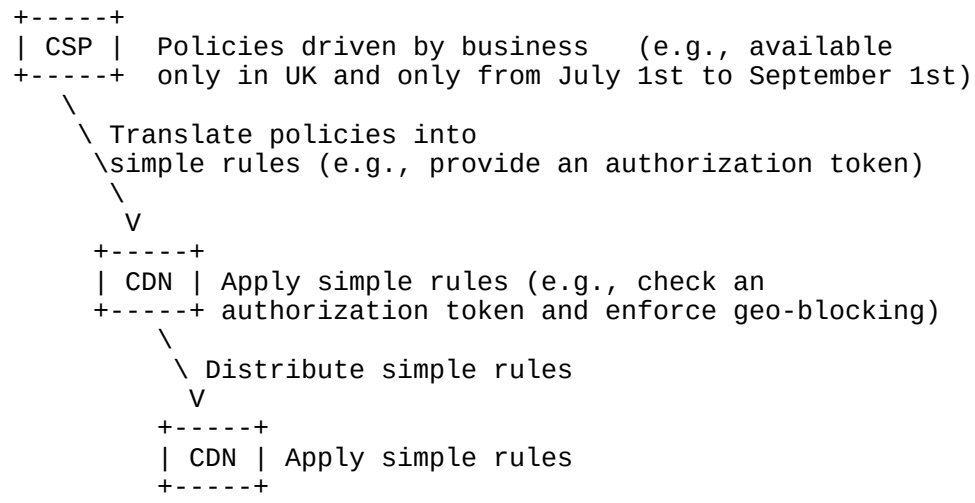


Figure 4

A.2. Secure Access

Many protocols exist for delivering content to End Users. CSPs may dictate a specific protocol or set of protocols which are acceptable for delivery of their content, especially in the case where a secured content transmission is required (e.g., must use HTTPS). CSPs may also perform per-request authentication/authorization decision and then have the CDNs enforce that decision (e.g., must validate URL signing, etc.).

A.3. Branding

Preserving the branding of the CSP throughout delivery is often important to the CSP. CSPs may desire to offer content services under their own name, even when the associated CDN service involves other CDN Providers. For instance, a CSP may desire to ensure that content is delivered with URIs appearing to the End Users under the CSP’s own domain name, even when the content delivery involves separate CDN Providers. The CSP may wish to prevent the delivery of its content by specific dCDNs that lack support for such branding preservation features.

Analogous cases exist when the uCDN wants to offer CDN services under its own branding even if dCDNs are involved. Similarly, a CDN Provider might wish to restrict the delivery delegation to a chain that preserves its brand visibility.

Authors' Addresses

Gilles Bertrand (editor)
France Telecom - Orange
38-40 rue du General Leclerc
Issy les Moulineaux, 92130
FR

Phone: +33 1 45 29 89 46
Email: gilles.bertrand@orange.com

Stephan Emile
France Telecom - Orange
2 avenue Pierre Marzin
Lannion F-22307
France

Email: emile.stephan@orange.com

Trevor Burbridge
BT
B54 Room 70, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: trevor.burbridge@bt.com

Philip Eardley
BT
B54 Room 77, Adastral Park, Martlesham
Ipswich, IP5 3RE
UK

Email: philip.eardley@bt.com

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

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

Grant Watson
Alcatel-Lucent (Velocix)
3 Ely Road
Milton, Cambridge CB24 6AA
UK

Email: gwatson@velocix.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 15, 2012

B. Niven-Jenkins
D. Ferguson
Velocix (Alcatel-Lucent)
G. Watson
BT

September 12, 2011

CDN Interconnect Metadata
draft-jenkins-cdni-metadata-00

Abstract

This document focuses on the CDNI Metadata interface, which enables the CDNI Metadata function in a Downstream CDN to obtain CDNI Metadata from an Upstream CDN so that the Downstream CDN can properly process and respond to Redirection Requests received over the CDNI Request Routing protocol and Request Routing and Content Requests received directly from User Agents.

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 March 15, 2012.

Copyright Notice

Table of Contents

1. Introduction

[I-D.jenkins-cdni-problem-statement] introduces the Problem scope for CDN Interconnection (CDNI) and lists the four categories of interfaces that may be used to compose a CDNI solution (Control, Metadata, Request Routing, Logging). [I-D.davie-cdni-framework] expands on the information provided in [I-D.jenkins-cdni-problem-statement] and describes each of the interfaces and the relationships between them in more detail.

This document focuses on the CDNI Metadata interface, which enables the CDNI Metadata function in a Downstream CDN to obtain CDNI Metadata from an Upstream CDN so that the Downstream CDN can properly process and respond to:

- Redirection Requests received over the CDNI Request Routing protocol.
- Request Routing and Content Requests received directly from User Agents.

Specifically this document proposes:

- A set of data objects that are used to describe the different aspects of CDNI Metadata along with a Data Model for CDNI Metadata that expresses the relationships between the CDNI Metadata objects (Section 2).
- An initial set of properties for the data objects in the CDNI Metadata data model (Section 3 through Section 5).
- A RESTful web service for the transfer of CDNI Metadata data objects (i.e. the CDNI Metadata interface) (Section 6).

Note: In order to make this document more accessible, it does not attempt to articulate all the CDNI Metadata that would be required to satisfy all CDNI use cases. Rather, a smaller set of properties are proposed. These should be sufficient to implement a minimal interconnect of basic HTTP delivery. Readers are encouraged to focus on the overall structure of the protocol rather than on the details or omission of particular properties. Additional properties may be added in future drafts or may be better placed in separate documents that focus on the CDNI Metadata required to interconnect delivery of individual end-user protocols.

1.1. Terminology

This document reuses the terminology defined in [I-D.jenkins-cdni-problem-statement].

2. CDNI Metadata Data Model

The CDNI Metadata interface specified in this document utilizes two types of Data Object:

- Addressable Data Objects, which are resources that may be retrieved via their own URIs.
- Embedded Data Objects, which are contained within a property of an Addressable Data Object.

Table 1 below defines the addressable data objects used by the CDNI Metadata interface.

Data Object	Description
SiteFeed	A SiteFeed object lists the Sites that may be delegated to the Downstream CDN.
Site	A Site object represents a customer-facing hostname that is used to serve content through including the rules for serving that content.
SelectionACL	A SelectionACL object restricts the Surrogates that can provide service for the associated Site to Surrogates in thre listed Locations.
DeliveryACL	A DeliveryACL object restricts the User Agent IP addresses that can access the associated Site or a URI pattern with the associated Site to User Agents in the listed Locations.
Location	A Location object represents a set of IP Address ranges.

Table 1: Content Distribution Metadata Data Objects

The only embedded data object defined is the DeliveryGlob object within a Site object, which describes the rules to apply for particular pattern based path matches within a Site.

The relationships between the different addressable CDNI Content Distribution Metadata objects are described in Figure 1 and Table 2 below and the properties of each object are described in Section 3.

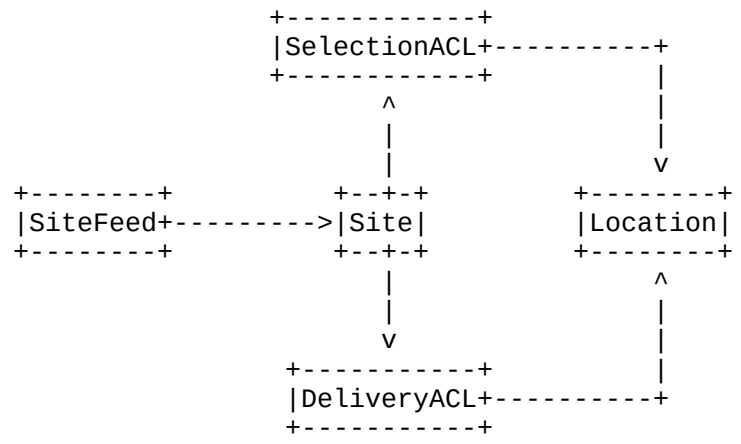


Figure 1: Relationships between CDNI Metadata Objects

Data Object	Objects it References
SiteFeed	0 or more Site objects.
Site	0 or 1 SelectionACL objects. 0 or 1 DeliveryACL objects.
SelectionACL	1 or more Location objects.
DeliveryACL	1 or more Location objects.
Location	None.

Table 2: Relationships between CDNI Metadata Objects

3. CDNI Metadata Addressable Data Object Descriptions

Each of the sub-sections below describes the properties associated with the data objects defined in Table 1.

The definition of each object is split into an ordered list of relationships and an unordered set of properties. Relationships are to other addressable data objects that are retrievable via the CDNI Metadata interface. The only exception is the DeliveryProtocol relationship which is to an 'opaque' URI representing the particular feature set of a delivery protocol defined in a specification and implemented in code.

Note: The names used for relationships (for example DeliveryProtocol) are illustrative and selected for readability and intuitive

understanding of their meaning. If retained, a future version of this document should list them out in the IANA considerations section and request they are registered in the Link registry.

3.1. SiteFeed

Relationship: Lists
Description: A Site that the Upstream CDN may delegate the delivery of to a Downstream CDN.
Allowed Target Types: Site
Cardinality: 0..*

Properties: None

3.2. Site

Relationship: DeliveryProtocol
Description: The Protocol to use for delivering this Site’s content.
Allowed Target Types: Protocol
Cardinality: 1

Relationship: RestrictsDelivery
Description: The DeliveryACL is used to restrict which End User IP addresses are allowed access to the content of this Site. If not present, there are no restrictions on which End Users may receive the associated content (i.e. any End User IP address can access the content).
Allowed Target Types: DeliveryACL
Cardinality: 0..1

Relationship: RestrictsSelection
Description: The SelectionACL is used to restrict which Surrogates are allowed to serve the content of this Site. If not present, there are no restrictions on which Surrogates may deliver the associated content (i.e. any server can serve the content).
Allowed Target Types: SelectionACL
Cardinality: 0..1

Property: active
Description: Whether delivery should be enabled for this Site.
Type: Boolean
Mandatory: No (default True)

Property: delivery.globs

```
         Description: Path specific rules.  First match applies.
         Type: List of DeliveryGlob
         Mandatory: No (default apply the properties defined in the Site
         object to all paths)

      Property: delivery.hostname
         Description: The customer facing hostname for this site.
         Type: Hostname
         Mandatory: Yes

      Property: delivery.query.remove
         Description: A list of query parameter names.  The listed query
         parameters must be removed before checking for presence in the
         cache or forwarding the request to the origin
         Type: List of Strings
         Mandatory: No (default pass full URI through)

      Property: origin.basic.active
         Description: Whether to use HTTP Basic authentication to the
         Origin.
         Type: Boolean
         Mandatory: No (default False)

      Property: origin.basic.password
         Description: HTTP Basic auth password.  Required if
         origin.basic.active is true.
         Type: String
         Mandatory: No (default no password)

      Property: origin.basic.username
         Description: HTTP Basic auth username.  Required if
         origin.basic.active is true.
         Type: String
         Mandatory: No (default no username)

      Property: origin.cookie.active
         Description: Whether to use cookie-based authentication when
         contacting the Origin server.
         Type: Boolean
         Mandatory: No (default False)

      Property: origin.cookie.value
         Description: Cookie value to be returned to origin for cookie-
         based authentication.  Required if origin.cookie.active is
         True.
         Type: String
```

Mandatory: No (default no cookie value)

Property: origin.endpoints
Description: Origins from which the Downstream CDN can acquire content. These are not necessarily the actual origin servers operated by the CSP but might be a set of Surrogates/servers in the Upstream CDN.
Type: Set of Endpoints
Mandatory: Yes

3.3. SelectionACL

Relationship: SelectionAllow
Description: Surrogates in the referenced Location are allowed to serve the content of this Site.
Allowed Target Types: Location
Cardinality: 0..*

Relationship: SelectionDeny
Description: Surrogates in the referenced Location are not allowed to serve the content of this Site.
Allowed Target Types: Location
Cardinality: 0..*

Properties: None

Note: The order of Relationships within a SelectionACL is the order in which the ACL MUST be processed. If a SelectionACL object does not contain any of the above relationships (i.e. the object is empty) the result is the equivalent of matching against a SelectionDeny entry (i.e. any server is allowed to serve the associated content). If the end of a SelectionACL is reached without matching any of its entries the result is the equivalent of matching against a SelectionDeny entry (i.e. no Surrogates are allowed to serve the content of the Site).

3.4. DeliveryACL

Relationship: DeliveryAllow
Description: User Agents in the referenced Location are allowed to receive the content of this Site/DeliveryGlob.
Allowed Target Types: Location
Cardinality: 0..*

Relationship: DeliveryDeny
Description: User Agents in the referenced Location are not allowed to receive the content of this Site/DeliveryGlob.

Allowed Target Types: Location
Cardinality: 0..*

Relationship: DeliveryAuth
Description: The referenced URI represents an Authorisation Server that must be queried to determine whether or not to allow clients to receive the content of this Site/DeliveryGlob.
Allowed Target Types: URI
Cardinality: 0..1

Properties: None

Note: The order of Relationships within a DeliveryACL is the order in which the ACL MUST be processed. If a DeliveryACL object does not contain any of the above relationships (i.e. the object is empty) the result is the equivalent of matching against a DeliveryDeny entry (i.e. any User Agent IP address is allowed to receive the associated content). If the end of an DeliveryACL is reached with matching any of its entries the result is the equivalent of matching against a DeliveryDeny entry (i.e. Delivery to the User Agent is not allowed).

3.5. Location

Relationships: None

Property: location.ip
Description: A set of IP Addresses.
Type: List of IPRange.
Mandatory: Yes

[Editor's Note: Location as specified above only supports the Class 1a names described in [I-D.jenkins-cdni-names]. Need to add support for Class 1b names to a later version.]

4. CDNI Metadata Embedded Data Objects Descriptions

Each of the sub-sections below describes the data objects that are embedded within one or more of the data objects described in Section 3.

As in the previous section, the definition of each object is split into its properties and its relationships. Relationships may be to other objects that are retrievable via the CDNI Metadata interface.

4.1. DeliveryGlob

Relationship: RestrictsDelivery
Description: The DeliveryACL is used to restrict which End User IP addresses are allowed access to the content matched by this DeliveryGlob. If not present, there are no restrictions on which End Users may receive the associated content (i.e. any End User IP address can access the content).
Allowed Target Types: DeliveryACL
Cardinality: 0..1

Property: pattern.string
Description: String to match against the requested path, i.e. a [RFC3986] path-absolute.
Type: Pattern.
Mandatory: Yes

Property: pattern.flags
Description: Flags to control the pattern match.
Type: PatternFlags.
Mandatory: No (default Case-sensitive infix matching)

Property: delivery.proxy
Description: If set to True then this pattern should be proxied and not cached.
Type: Boolean.
Mandatory: No (default False)

5. CDNI Metadata Simple Data Type Descriptions

This section describes the simpler data types that are used for properties of Addressable Data Objects and Embedded Data Objects.

5.1. Protocol

This type only appears in links. Links with this type are not machine readable but rather represent particular feature sets of a protocol defined in a specification and implemented in code. The URI contained in the link needs to be defined for each delivery protocol with an associated interoperable feature set.

The following examples are illustrative:

- http://url.cdni.ietf.example/protocol/delivery/http/rfcABCD
- http://url.cdni.ietf.example/protocol/delivery/rtmp/rfcEFGH

- http://url.vendorY.ietf.example/protocol/delivery/rtmp/releaseP.Q

[Editor's Note: It may be more appropriate to use the 'tag' URI scheme [RFC4151] for these URIs.]

5.2. Endpoint

A hostname (with optional port) or an IP address (with optional port).

Note: Client implementations MUST support IPv4 addresses encoded as specified by the 'IPv4address' rule in Section 3.2.2 of [RFC3986] and MUST support all IPv6 address formats specified in [RFC4291]. Server implementations SHOULD use IPv6 address formats specified in [RFC5952].

5.3. IPRange

One of:

- A range of consecutive IP addresses (IPv4 or IPv6) expressed as Address1-Address2 which does not have to be to power of two aligned, for example the range 192.0.2.1-192.0.2.10 is valid. The first Address in the range MUST be 'lower' than the final address in the range.
- A valid IP subnet (IPv4 or IPv6) expressed using CIDR notation.
- A single IP address (IPv4 or IPv6).

Note: Client implementations MUST support IPv4 addresses encoded as specified by the 'IPv4address' rule in Section 3.2.2 of [RFC3986] and MUST support all IPv6 address formats specified in [RFC4291]. Server implementations SHOULD use IPv6 address formats specified in [RFC5952].

5.4. Pattern

A pattern for string matching. The string may contain the wildcards * and ?:

- * matches any sequence of characters (including the empty string).
- ? matches exactly one character.

Escaping: The three literals \ , * and ? should be escaped as \\, * and \?

5.5. PatternFlags

A set of flags indicating how a pattern match is made. The flags are:

- Case-insensitive - Perform a case insensitive match (absence indicates case-sensitive match).
- Prefix - Match against the start of the string (absence indicates that a match may start anywhere in the string).
- Suffix - Match against the end of the string (absence indicates that a match may end anywhere in the string).

Absence of both Prefix and Suffix results in a match against any part of the string (infix).

5.6. URI

A URI as specified in [RFC3986].

6. CDNI Metadata interface

This section specifies an interface to enable a Downstream CDN to retrieve CDNI Metadata objects from an Upstream CDN.

The CDNI Metadata interface is built on the principles of RESTful web services, in particular the use of hypermedia as the engine of application state, and follows patterns established in The Atom Syndication Format [RFC4287]. This means that requests and responses over the interface are built around the transfer of representations of hyperlinked resources. A resource in the context of the CDNI Metadata interface is an Addressable Object in the Data Model (as described in Section 2 through Section 3).

Requests are made over HTTP. The HTTP Method defines the operation the request would like to perform. Servers implementing the CDNI Metadata interface MUST support the HTTP GET and HEAD methods. The corresponding HTTP Response returns the status of the operation in the HTTP Status Code and returns the current representation of the resource (if appropriate) in the Response Body. HTTP Responses from servers implementing the CDNI Metadata interface that contain a response body SHOULD include an ETag to enable validation of cached versions of returned resources.

The CDNI Metadata interface specified in this document is a read-only interface. Therefore support for other HTTP methods such as PUT, POST and DELETE etc. is not specified. Server implementations of this interface SHOULD reject all methods other than GET and HEAD.

The only representation specified in this document is JSON.

The interface can be used by a Downstream CDN to retrieve CDNI Metadata objects either dynamically as required by the Downstream CDN to process received requests (for example in response to receiving a CDNI Request Routing request from an Upstream CDN or in response to receiving a request for content from a User Agent) or in advance of being required.

In the general case a CDNI Metadata server makes each instance of an addressable CDNI Metadata object available via a unique URI that returns a representation of that instance of that CDNI Metadata object. When an object needs to reference another addressable CDNI Metadata object (for example a Site object referencing a DeliveryACL object) it does so by including a link to the referenced object.

The URI for the SiteFeed object needs to be either discovered by or configured in the downstream CDN. All other objects/resources are then discoverable from the SiteFeed object by following the links in the SiteFeed object and the referenced Site objects. CDNI Metadata servers are therefore free to assign whatever structure they desire to the URIs for CDNI Metadata objects and CDNI Metadata clients MUST NOT make any assumptions regarding the structure of CDNI Metadata URIs or the mapping between CDNI Metadata objects and their associated URIs. Therefore any URIs present in the examples below are purely illustrative and are not intended impose a definitive structure on CDNI Metadata interface implementations.

As the CDNI Metadata interface builds on top of HTTP, CDNI Metadata servers may make use of any HTTP feature when implementing the CDNI Metadata interface, for example a CDNI Metadata server may make use of HTTP’s caching mechanisms to indicate that the returned response/ representation can be reused without re-contacting the CDNI Metadata server.

6.1. MIME Media Types

Table 3 lists the MIME Media Type for each object (resource) that is retrievable through the CDNI Metadata interface as well as the MIME Media Type for the DeliveryProtocol relationship.

Note: A prefix of "vnd.cdni" is used, however it is expected that a more appropriate prefix will be used if this document is accepted by the CDNI WG.

Data Object	MIME Media Type
SiteFeed	application/vnd.cdni.metadata.site.feed+json
Site	application/vnd.cdni.metadata.site+json
SelectionACL	application/ vnd.cdni.metadata.acl.selection+json
DeliveryACL	application/ vnd.cdni.metadata.acl.delivery+json
Location	application/vnd.cdni.metadata.location+json
DeliveryProtocol	application/vnd.cdni.metadata.protocol+json

Table 3: MIME Media Types for CDNI Metadata resources

6.2. JSON Encoding of Objects

The "base" encoding for a CDNI Metadata object is a JSON object containing a dictionary of (key,value) pairs where the keys are the property names and the values are the associated property values.

The keys of the dictionary are the names of the properties associated with the object and are therefore dependent on the specific object being encoded (i.e. dependent on the MIME Media Type of the returned resource). Likewise, the values associated with each key are dependent on the specific object being encoded (i.e. dependent on the MIME Media Type of the returned resource).

Dictionary keys in JSON are case sensitive and therefore any dictionary key defined by this document (for example the names of CDNI Metadata object properties) MUST always be represented in lowercase.

In addition to the properties of the object, the following three additional keys defined below may be present.

Key: base
Description: Provides a prefix for any relative URLs in the object. This is similar to the XML base tag [XML-BASE]. If absent, all URLs in the remainder of the document must be absolute URLs.
Type: URI
Mandatory: No

Key: links
Description: The relationships of this object to other addressable objects.

```
         Type: List of Relationships.
         Mandatory: Yes
      Key: inline
         Description: Dictionary containing additional objects that are
         inlined within this object.  The keys in the dictionary are
         then then used to generate URI fragments which are used to
         refer to inlined objects and the value is the Object itself.
         Type: Dictionary of Objects.
         Mandatory: No
```

Example SiteFeed Object:

```
{
  "base": "http://metadata.cdni.example.com/sites/",
  "links": [
    {
      "title": "videos.example.net",
      "href": "example1",
      "rel": "Lists",
      "type": "application/vnd.cdni.metadata.site+json"
    },
    {
      "title": "trailers.example.net",
      "href": "http://metadata2.cdni.example.com/sites/example2",
      "rel": "Lists",
      "type": "application/vnd.cdni.metadata.site+json"
    }
  ],
}
```

6.3. JSON Encoding of Embedded Types

6.3.1. PatternFlags

JSON: A number calculated by adding together the values associated with each flag that is set:

- 1 - Case-insensitive
- 2 - Prefix
- 4 - Suffix

Example of case-insensitive prefix match:

```
"pattern.flags": 3
```

6.3.2. Protocol

TBD

6.3.3. Relationship / Link

JSON: A dictionary with the following keys:

- href - The URI of the of the addressable object being referenced.
- rel - The Relationship between the referring object and the object it is referencing.
- type - The MIME Media Type of the referenced object. See Section 6.1 for the MIME Media Types of objects specified in this document.
- title - An title for the for the Relationship/link. For the "Lists" Relationships contained in a SiteFeed object the title key MUST be present and MUST be the value of delivery.hostname for the referenced Site object. For all other Relationships/links the title key is optional.

Note: The above structure follows the pattern of atom:link in [RFC4287].

Example Relationship to a CDNI Metadata location within a SelectionACL object:

```
{
  "href": "http://metadata.cdni.example.com/locations/everywhere",
  "rel": "SelectionAllow",
  "type": "application/vnd.cdni.metadata.location+json"
}
```

6.4. Retrieval of CDNI Metadata resources

In the general case a CDNI Metadata server makes each instance of an addressable CDNI Metadata object available via a unique URI and therefore in order to retrieve CDNI Metadata, a CDNI Metadata client first makes a HTTP GET request for the URI of the SiteFeed which provides the CDNI Metadata client with a list of Sites (along with their public facing hostnames) that the upstream CDN may delegate to the downstream CDN.

In order to retrieve the CDNI Metadata for a particular Site the CDNI Metadata client processes the received SiteFeed object and finds the corresponding entry (using the title key to match against the required public facing hostname if required) in the SiteFeed for the Site it requires and can then can make a GET request for the URI specified in the href key of that Site’s entry in the SiteFeed.

In order to retrieve the SelectionACLs and DeliveryACLs associated with that Site the CDNI Metadata client processes the received Site object (as well as any DeliveryGlob objects embedded in the Site's delivery.globs key) and can then can make a GET request for the URIs specified in the href key of links within that Site or its DeliveryGlobs that have a Relationship of RestrictsSelection and RestrictsDelivery respectively.

Finally in order to retrieve the Locations associated with each ACL the CDNI Metadata client processes the received ACL object(s) and can then can make a GET request for the URIs specified in the href key of links within that ACL that have a Relationship of Location.

6.4.1. Bulk Retrieval of CDNI Metadata resources

In addition to the general case where a CDNI Metadata server makes each instance of an addressable CDNI Metadata object available via a unique URI, in response to a request for a CDNI Metadata object a CDNI Metadata Servers MAY include one or more of the addressable objects referenced by the requested object inside the inline dictionary of the referenced objects.

Inlined objects are referenced using a link in the normal way with the exception that the href component of the link begins with # and follows the fragment resolution protocol defined in [I-D.zyp-json-schema].

Use of the inline dictionary to include multiple addressable objects has the advantage of reducing the number of requests a CDNI Metadata client needs to make (and therefore reduces the additional latency introduced by making multiple requests) in order to retrieve all the CDNI Metadata it requires to process CDNI Request Routing requests and User Agent requests for content.

However use of the inline dictionary has the disadvantage that the cacheability of any inlined objects are tied to the cacheability of the object they are inlined in. Some objects (for example Locations) may not change very often and therefore could be cached for a longer period of time than the objects that reference them. Or an object (for example a SelectionACL) may be referenced by multiple other objects and benefit from being cached separately in order to reduce the size of the responses to requests for objects that reference it.

Example of an inline dictionary containing a DeliveryACL object:

```
"inline": {
  "deliveryeverywhereacl": {
    "links": [
      {
        "href":
          "http://metadata.cdni.example.com/locations/everywhere",
        "rel": "DeliveryAllow",
        "type": "application/vnd.cdni.metadata.location+json"
      }
    ]
  }
}
```

Example of an inline dictionary containing a DeliveryACL object where the Location referenced by the DeliveryACL is also inlined:

```
"inline": {
  "deliveryeverywhereacl": {
    "links": [
      {
        "href": "#/inline/everywhere",
        "rel": "DeliveryAllow",
        "type": "application/vnd.cdni.metadata.location+json"
      }
    ]
  },
  "everywhere": {
    "location.ip": ["0.0.0.0", "::/0"]
  }
}
```

Note: An alternative to using an inline key as described above would be for the CDNI Metadata server to return a multipart/mixed response. Using a multipart/mixed response would have the advantage that inlined objects would not have to be tied to the cacheability of the object they are inlined in (as they could have their own Cache-Control headers) and could be cached separately from the object they are inlined with (as they could have their own Location header). However, using a multipart/mixed response would have the disadvantage of requiring more complex processing by the CDNI Metadata client.

6.5. Examples

The following sections provide examples of different CDNI Metadata objects encoded as JSON.

6.5.1. SiteFeed

Example SiteFeed:

```
{
  "base": "http://metadata.cdni.example.com/sites/",
  "links": [
    {
      "title": "videos.example.net",
      "href": "example1",
      "rel": "Lists",
      "type": "application/vnd.cdni.metadata.site+json"
    },
    {
      "title": "trailers.example.net",
      "href": "http://metadata2.cdni.example.com/sites/example2",
      "rel": "Lists",
      "type": "application/vnd.cdni.metadata.site+json"
    }
  ],
}
```

6.5.2. Site

Example Site with inlined objects:

```
{
  "base": "http://metadata.cdni.example.com/",
  "links": [
    {
      "href": "http://url.cdni.ietf.org/protocol/delivery/http/1.1",
      "rel": "DeliveryProtocol",
      "type": "application/vnd.cdni.metadata.protocol+json"
    },
    {
      "href": "#/inline/delivereverywhereacl",
      "rel": "RestrictsDelivery",
      "type": "application/vnd.cdni.acl.delivery+json"
    },
    {
      "href": "#/inline/servefromanywhereacl",
      "rel": "RestrictsSelection",
      "type": "application/vnd.cdni.acl.selection+json"
    }
  ],
  "active": "true",
  "delivery.hostname": "videos.example.net"
  "origin.hostnames": ["origin.videos.example.net.cdni.example.com",
```

```
      "origin.example.net"],
    "delivery.globs": [
      {
        "links": [
          {
            "href": "#/inline/qaonlyacl",
            "rel": "RestrictsDelivery",
            "type": "application/vnd.cdni.acl.delivery+json"
          }
        ],
        "pattern.string": "*/test_files/*",
        "pattern.flags": 1,
        "delivery.proxy": false
      }
    ],
    "inline": {
      "deliveryeverywhereacl": {
        "links": [
          {
            "href": "#/inline/everywhere",
            "rel": "DeliveryAllow",
            "type": "application/vnd.cdni.metadata.location+json"
          }
        ]
      },
      "servefromanywhereacl": {
        "links": [
          {
            "href": "#/inline/everywhere",
            "rel": "SelectionAllow",
            "type": "application/vnd.cdni.metadata.location+json"
          }
        ]
      },
      "qaonlyacl": {
        "links": [
          {
            "href": "#/inline/qa1",
            "rel": "DeliveryAllow",
            "type": "application/vnd.cdni.metadata.location+json"
          },
          {
            "href": "#/inline/qa2",
            "rel": "DeliveryAllow",
            "type": "application/vnd.cdni.metadata.location+json"
          },
          {
            "href": "#/inline/everywhere",
```

```
            "rel": "DeliveryDeny",
            "type": "application/vnd.cdni.metadata.location+json"
          }
        ]
      },
      "everywhere": {
        "location.ip": ["0.0.0.0", "::/0"]
      },
      "qa1": {
        "location.ip": ["192.0.2.1-192.0.2.10"]
      },
      "qa2": {
        "location.ip": ["198.51.100.1-198.51.100.15",
          "198.51.100.200-198.51.100.254"]
      }
    }
  }
```

7. IANA Considerations

TBD.

8. Security Considerations

TBD.

9. Acknowledgements

TBD.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

10.2. Informative References

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[I-D.jenkins-cdni-names]
Niven-Jenkins, B., "Thoughts on Naming and Referencing of Data Objects within Content Distribution Network Interconnection (CDNI) solutions", draft-jenkins-cdni-names-00 (work in progress), February 2011.

[I-D.jenkins-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-jenkins-cdni-problem-statement-02 (work in progress), March 2011.

[I-D.lefaucheur-cdni-requirements]
Leung, K., Lee, Y., Faucheur, F., Viveganandhan, M., and G. Watson, "Content Distribution Network Interconnection (CDNI) Requirements", draft-lefaucheur-cdni-requirements-02 (work in progress), July 2011.

[I-D.zyp-json-schema]
Zyp, K. and G. Court, "A JSON Media Type for Describing the Structure and Meaning of JSON Documents", draft-zyp-json-schema-03 (work in progress), November 2010.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4151] Kindberg, T. and S. Hawke, "The 'tag' URI Scheme", RFC 4151, October 2005.

[RFC4287] Nottingham, M., Ed. and R. Sayre, Ed., "The Atom Syndication Format", RFC 4287, December 2005.

[XML-BASE]
Marsh, J., Ed. and R. Tobin, Ed., "XML Base (Second Edition) - http://www.w3.org/TR/xmlbase/", January 2009.

Appendix A. Relationship to the CDNI Requirements

Section 6 of [I-D.lefaucheur-cdni-requirements] lists the requirements for the CDNI Metadata Distribution interface. This section outlines which of those requirements are met by the CDNI Metadata interface specified in this document.

Requirements R49 through R57 (inclusive) and R61 through R63 (inclusive) are directly met by the interface specified in this document.

Requirements R59 and R60 can be trivally met by defining additional properties for the CDNI Metadata objects defined in this document.

It is the opinion of the authors that requirement R58 is better handled at Request Routing time by the CDNI Request Routing interface, rather than directly being met by the CDNI Metadata interface.

Authors' Addresses

Ben Niven-Jenkins
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK

Email: ben@velocix.com

David Ferguson
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK

Email: david@velocix.com

Grant Watson
BT
pp GDC 1 PP14, Orion Building, Adastral Park
Martlesham, Ipswich IP5 3RE
UK

Email: grant.watson@bt.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 17, 2013

K. Ma
Azuki Systems, Inc.

July 16, 2012

Content Distribution Network Interconnection (CDNI) Metadata Interface
draft-ma-cdni-metadata-03

Abstract

Content publishers (CPs) often use multiple Content Delivery Networks (CDNs) to deliver content to consumers. Though existing interactions between CPs and individual CDNs are beyond the scope of CDN interconnection (CDNI), it is important to understand the management capabilities and features available with existing non-interconnected multi-CDN deployments. Before migrating to CDNI, CPs must first assess the suitability of CDNI as a replacement for their existing non-interconnected multi-CDN deployments. CDN feature configuration and capability advertisement and enforcement is likely to occur through the CDNI metadata interface (MI). This document describes an approach to implementing the CDNI MI through the use of an extensible metadata model and a light-weight HTTP-based API.

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 January 17, 2013.

Copyright Notice

Table of Contents

1. Introduction

The use cases described in the CDNI use case document [I-D.ietf-cdni-use-cases] provide motivational use cases for CDN interconnection (CDNI). They describe reasons and situations where CDNI provides a benefit to CDN vendors as well as content service providers (CSPs). Additional use cases exist which describe how CDNs are used today, however, these use cases often involve specific features (e.g., customized content transformations, content security, client authentication and filtering, content acquisition optimization and redundancy, etc.) which are beyond the scope of CDNI. Though the features themselves are not relevant to CDNI, the ability to support those features or enforce policies related to those features in a generic and extensible manner should be considered when designing CDNI interfaces. The ability to support feature parity with existing deployment models (i.e., non-CDNI-based CDN federation) may help to remove barriers to CDNI adoption.

Though certain interfaces are out of scope of CDNI, e.g.:

- upstream CDN (uCDN) configuration by the CP
- uCDN content acquisition
- uCDN content delivery
- downstream CDN (dCDN) content acquisition
- end user (EU) content acquisition
- third party workflow management
- third party request routing

An awareness of these interfaces and an understanding of the restrictions which they may impose on CDNI request routing is useful for understanding the needs of the CDNI metadata interface (MI). As described in the "Dynamic CDNI Metadata Acquisition Example" section in the CDNI framework document [I-D.davie-cdni-framework], upon receiving a request routing interface (RRI) request, the MI MAY be used to retrieve metadata that is "considered" before responding to the RRI request. To that end, the MI MUST define a deterministic method for handling metadata processing. Though the definition and interpretation of any individual piece of metadata is beyond the scope of CDNI, a well-defined method for how to respond to a RRI request when any unknown metadata value is encountered MUST be supported.

This document describes a simple data model for representing CDNI metadata and a simple protocol for creating and retrieving CDNI metadata in an opaque manner. The term opaque, in this case, should be understood to mean: without understanding the underlying meaning or interpretation of the metadata being represented. The metadata model and retrieval protocol SHOULD be completely independent of the definition of individual metadata values. The metadata model and retrieval protocol MUST also define default behaviors for dealing with metadata processing errors. The document defines a list of metadata which are likely applicable to a broad range of CDNI deployments. The document also provides a separate list of metadata which are likely to be desirable to content publishers (CPs). This document is not intended to suggest that any additional interfaces or requirements are needed beyond those already specified in the CDNI requirements document [I-D.ietf-cdni-requirements], nor is this document intended to suggest that any out of scope interfaces or content publisher feature functionality should be brought into scope. The metadata examples provided are intended only to illustrate possible features that interconnected CDNs may wish to support and the extensibility of the metadata model to handle those situations.

1.1. Terminology

[Ed. insert terminology reference]

1.2. Abbreviations

- CDN: Content Distribution Network
- uCDN: Upstream Content Distribution Network
- dCDN: Downstream Content Distribution Network
- CDNI: Content Distribution Network Interconnection
- CP: Content Publisher
- CSP: Content Service Provider
- EU: End User
- NSP: Network Service Provider
- RRI: Request Routing Interface
- MI: Metadata Interface

o CI: Control Interface

2. CDNI Metadata Data Model

The simple data model is shown in Figure 1 below. It includes a top level Domain object which describes the site(s) to which metadata is associated. The term site, in this case, should be understood to mean a collection of related content assets accessed through a single portal or Web-site. The Domain is associated with zero or more opaque Metadata objects. Each Metadata object is associated with one or more Base Address objects. The Metadata objects are each associated with a URI extension, applicable to any of the associated Base Addresses. A combination of Base Address and URI prefix matching is used identify Metadata to allow for hierarchical associations between individual Metadata and sets of content items. Each Domain is also associated with one or more Agent objects. Agents represent entities which require access to metadata (e.g., CPs, uCDNs, dCDNs, or local operators). An Agent is associated with each Metadata entry allowing different Metadata values to be returned to different Agents.

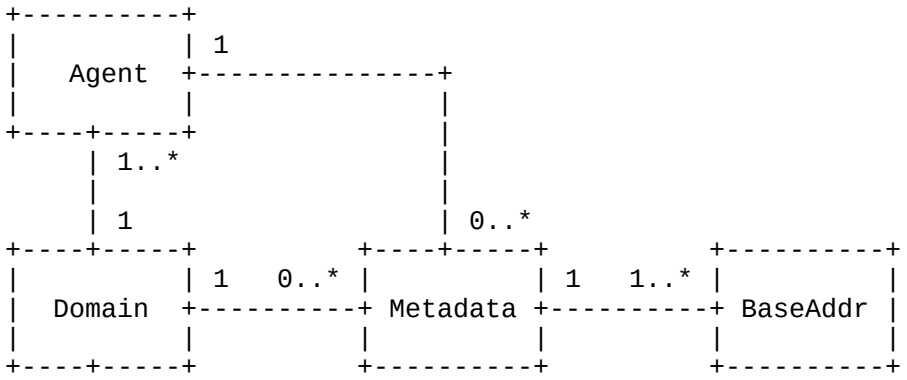


Figure 1: CDNI Metadata Data Model

Note: The data model described above provides the basic components required for distributing Metadata and implementing the CDNI MI. The specific semantics of individual pieces of metadata are abstracted to allow for opaque distribution of metadata. Not all of the information described need be distributed through the MI. Some information (e.g., Domains and Agents) may be necessary for the MI to function, but MAY be negotiated or implemented out-of-band. They could be configured either by the CDN as part of a non-CDNI process, or through the CDNI control interface (CI) bootstrapping process, or using the MI APIs described herein. The MI APIs may also be used by

CDNs, internally, to configure themselves. The complete data model and full set of APIs are provided as part of a holistic MI description.

The following sections describe an example implementation of the metadata scheme described above using a standard SQL database.

2.1. Domain Table

The Domain object contains basic information related to the site being described. The example shown contains a primary key index and a unique name for the site. An OPTIONAL site description (e.g., a textual description of the site and its content) and site provider (e.g., the name of the CP or CSP which owns the content) information is also included.

```
CREATE TABLE "domain" ("domain_id" serial primary key,
                       "name" character varying(255) NOT NULL,
                       "provider" character varying(255),
                       "description" character varying(4095));
CREATE UNIQUE INDEX index_domain ON domain (name);
```

The Domain is the central object for binding Metadata. The example Domain shown below highlights the descriptive nature of the Domain object:

```
domain_id: 1
name: acme
provider: acme rocket-powered products, inc
description: fine purveyors of high quality anvils, rubber bands,
             bird seed, and rocket-powered footwear.
```

2.2. Base Address Table

The Base Address object contains basic hostname and base URI information related to the site being accessed. The example shown requires a primary key index, a string containing the hostname and base URI, and a foreign key reference to the Metadata to which this Base Address is associated. A uniqueness constraint is imposed on baseaddr/metadata_id pairs to prevent duplicate Base Address entries for a given Metadata.

```
CREATE TABLE "baseaddr" ("baseaddr_id" serial primary key,
                         "baseaddr" character varying(255) NOT NULL,
                         "metadata_id" integer NOT NULL);
CREATE UNIQUE INDEX index_baseaddr ON baseaddr (baseaddr,
                                                metadata_id);
```

Base Address Table Definition

The Base Address objects allows multiple hostname and base URI pairs to be associated with each Metadata object denoting the list of Base Addresses through which content within the Domain may be accessed. There are many cases where different Base Addresses are used to access the same content, e.g.:

- internal vs. external addresses: content may be accessible via both internal 10-net IP addresses and their associated DNS addresses and base URIs, as well as publicly routable external IP addresses and their associated DNS addresses and base URIs, where all of the addresses point to the same content servers and the base URIs are mapped to the same base directories,

- service white-labeling: multiple CSPs may provide access to the same content through different branded services where each branded service has its own DNS address and/or base URI, but all of the services point to the same content, or

- analytics partitioning: redirects from other sites may use different DNS addresses and/or base URIs, so that they may be easily accounted for, while still pointing at the same content.

The example Base Addresses shown below represent two DNS addresses through which content may be accessed as well as an internal IP address which may be used for staging:

```
baseaddr_id: 1
baseaddr: wile.e.coyote.acme.com
metadata_id: 1

baseaddr_id: 2
baseaddr: road.runner.acme.com
metadata_id: 1

baseaddr_id: 3
baseaddr: 10.10.10.10/meemeep
metadata_id: 1
```

Note: The exact schema described above may result in heavy duplication of Base Addresses. It is presented as an example for its simplicity, however, it may be optimized by using other table joining implementation schemes.

2.2.1. Hierarchical Base Addresses

In order to support hierarchical Base Addresses, the wildcard '*.' SHOULD be allowed as the first part of DNS-type Base Addresses. The wildcard does not make sense at the beginning of IP Address-type Base Addresses. Though a wildcard at the end of IP Address-type Base Addresses would make more sense, support for IP Address-type Base Addresses is OPTIONAL. The wildcard signifies the applicability of the associated Metadata value to all Base Addresses which match the address suffix.

The following two Base Addresses condense the previous example by allowing all acme.com DNS addresses:

```
baseaddr_id: 1
baseaddr: *.acme.com
metadata_id: 1

baseaddr_id: 2
baseaddr: 10.10.10.10/meemeep
metadata_id: 1
```

Note: There is no explicit enforcement that Base Addresses associated with a given piece of Metadata not overlap, however, for performance reasons, Base Addresses associated with a given piece of Metadata SHOULD NOT be allowed to overlap.

2.3. Agent Table

The Agent object contains basic information for authenticating entities which require access to Metadata. The example shown contains a primary key index, a string containing the username, an OPTIONAL string containing the password (possibly hashed or encrypted), a boolean value for differentiating between full read/write access (e.g., for uCDNs) and read only access (e.g., for dCDNs), and a foreign key reference to the Domain to which this Agent is associated. A uniqueness constraint is imposed on username/domain_id pairs to prevent duplicate Agent entries for a given Domain.

```
CREATE TABLE "agent" ("agent_id" serial primary key,
                      "username" character varying(255) NOT NULL,
                      "password" character varying(255),
                      "read_only" boolean DEFAULT false NOT NULL,
                      "domain_id" integer NOT NULL);
CREATE UNIQUE INDEX index_agent ON agent (username, domain_id);
```

Agent Table Definition

Note: The password field is included to support the HTTP authentication described in the API sections, however, if alternate authentication schemes are used, the password may not be necessary.

The Agent objects manage Metadata access rights. The Agent functionality as described attempts to address two issues:

- security concerns: where unauthorized injection or deletion of Metadata may alter the functionality of a content service and MUST be prevented, as described in the Security Considerations section, and
- customization requirements: where retrieval of certain metadata may require different responses depending on the Agent who is accessing the Metadata (e.g., with multiple and/or cascaded dCDNs).

Note: Though both of the above issues could be addressed through means outside of the MI, or through a means common to all of the CDNI interfaces, the Agent serves the purpose of addressing these needs within the context of the MI, in lieu of a consensus alternative and as per the CDNI framework document [I-D.davie-cdni-framework].

The example Agents shown below represent a uCDN Agent with write privileges and two dCDN Agents with read-only permissions:

```
agent_id: 1
username: ucdn
password: xxx
read_only: false
domain_id: 1

agent_id: 2
username: dcdn1
password: yyy
read_only: true
domain_id: 1

agent_id: 3
username: dcdn2
password: zzz
read_only: true
domain_id: 1
```

2.4. Metadata Table

The Metadata object contains the actual individual pieces of metadata for the site being described. The example shown contains a primary key index, a string containing the URI(s) to which the metadata applies, a name/value pair of strings which represent the name and value of the Metadata, respectively, as well as a boolean value stating whether or not the given Metadata must be enforced. An OPTIONAL priority value is included for creating order lists of values for a given named Metadata. An OPTIONAL ttl value and timeout field are included to support metadata invalidation. The table also contains a foreign key reference to the Domain to which this Metadata is associated and a foreign key reference to the Agent to whom this Metadata is intended. A compound uniqueness constraint is also applied to each uri/name/priority/domain_id/agent_id tuple to prevent a given Metadata from being ambiguously applied multiple times to the same URI in a given Domain for a given Agent.

```
CREATE TABLE "metadata" ("metadata_id" serial primary key,
                         "uri" character varying(4095) NOT NULL,
                         "name" character varying(511) NOT NULL,
                         "value" character varying(65535) NOT NULL,
                         "must_enforce" boolean DEFAULT true NOT NULL,
                         "priority" integer DEFAULT 0 NOT NULL,
                         "ttl" integer DEFAULT 0 NOT NULL,
                         "timeout" timestamp without time zone,
                         "domain_id" integer NOT NULL,
                         "agent_id" integer NOT NULL);
CREATE UNIQUE INDEX index_metadata ON metadata (uri, name, priority,
                                                domain_id, agent_id);
```

The name/value pair is represented as simple opaque strings. The MI does not require and understanding of the semantics or inherent meaning of Metadata names or values to distribute the Metadata. Though, each piece of Metadata MUST have a defined set of semantics in order to be enforced, distributing the Metadata and determining whether or not the Metadata is supported does not require any understanding of the Metadata semantics, but rather, only an ability to identify supported Metadata by their name is REQUIRED. Metadata names SHOULD be properly defined and registered, and any implied functionality SHOULD be agreed upon and documented. A base set of CDNI Metadata is provided in the Metadata Definitions Section.

The intent of the must_enforce boolean is to identify Metadata that MUST be enforced by all CDNs. If a CDN is unable to understand or is unable to comply with the Metadata, it MUST NOT deliver the content being requested. For dCDNs, the must_enforce flag defines how to respond to MI and RRI requests when unknown or unsupported Metadata

is encountered. If Metadata is marked as must_enforce, then the dCDN MUST NOT accept any RRI request if it is unable to enforce that piece of Metadata (e.g., the named Metadata is not supported, the Metadata value is invalid, or the Metadata value is not supported). If the MI request resulted from a "recursive" RRI request, then the dCDN MUST return an error to the uCDN. If the MI request resulted from an "iterative" RRI request, then the dCDN MUST respond with a 403 Forbidden status code to the EU and report the failure to the uCDN.

In the case of cascaded CDN deployments, though a given CDN may not be able to enforce a given piece of Metadata, other CDNs further down stream may be able to enforce that Metadata. When a Metadata rejection occurs, the CDN SHOULD still store the Metadata so that it can be provided to other dCDNs.

The OPTIONAL priority value is provided to allow configuration of ordered Metadata lists. When specifying multiple values for a given named Metadata, each value MUST be specified with a unique priority value. The explicit priority value enforces a deterministic ordering across MI implementations.

The OPTIONAL ttl value is provided to allow configuration of a Metadata TTLs. If the ttl is specified, it MUST be specified in seconds and the timeout field SHOULD be populated by the local MI processor and used internally, to prevent the need for clock synchronization between MI processors.

The association of each Metadata to an Agent allows different Agents to retrieve different Metadata values for a given URI in the given Domain. This is intended to allow CDNs to separate upstream Metadata from downstream Metadata (e.g., a uCDN content acquisition URL may point to a CP origin, however, the content acquisition URL that the dCDN retrieves from the uCDN may point at a surrogate in the uCDN; likewise the content acquisition URLs for different dCDNs may point at different surrogates in the uCDN). Though this information could be hidden within a CDN’s implementation, the security aspects related to deterministically associating an authenticated Agent with the proper metadata should be considered as part of the MI. Explicitly representing this in the data model reduces ambiguity in Metadata retrieval.

2.4.1. Hierarchical Metadata

In order to support hierarchical metadata, ’/*’ SHOULD be allowed as the last part of the URI hierarchy, signifying the application of this Metadata value to all URIs which match this URI prefix. If multiple Metadata are defined with overlapping prefixes, the URI with the longest prefix match MUST be used. The uniqueness constraint on

the uri/name/priority/domain_id tuple allows for unambiguous resolution of Metadata priority.

Note: The wildcard is only supported at the end of the URI string to provide a well-defined ordering for URI prefixes (i.e., longest prefix matching). Use of generalized regular expression matching requires ordering rules to ensure deterministically coherent results across multiple MI implementations. It is assumed that the URI path extensions (beyond the base paths provided in the Base Address) for content will be the same across CDNs. Any CDN specific URL rewrites MUST only affect the Base Address portion of the URL as defined in the Base Address.

Note: It is often desirable to separate specific types of files which may live in the same directory (e.g., .m3u8 vs .ts). Wildcard support in the URI support for file extension differentiation, i.e., '/*[.extension]', is OPTIONAL.

Given the following four Metadata objects, the value of color is defined five times, for three different URIs, all within the same Domain, but for different Agents:

```
metadata_id: 1
uri: /*
name: color
value: blue
must_enforce: false
priority: 0
ttl: 0
domain_id: 1
agent_id: 2

metadata_id: 2
uri: /*
name: color
value: gold
must_enforce: false
priority: 0
ttl: 0
domain_id: 1
agent_id: 1

metadata_id: 3
uri: /*
name: color
value: blue
must_enforce: false
priority: 1
```

```
ttl: 0
domain_id: 1
agent_id: 1

metadata_id: 4
uri: /grass/*
name: color
value: brown
must_enforce: false
priority: 0
ttl: 0
domain_id: 1
agent_id: 2

metadata_id: 5
uri: /grass/on/the/other/side/*
name: color
value: green
must_enforce: false
priority: 0
ttl: 0
domain_id: 1
agent_id: 2
```

The default value for the color metadata (signified by the all encompassing URI "/*") is blue for the dCDN Agent and gold for the uCDN Agent, though the default color may be blue for the uCDN as well (as signified by the lower priority alternate color value). Alternate colors are associated with requests from the dCDN Agent for URIs that begin with "/grass". By default "/grass" has a color of brown, except when requesting "/grass/on/the/other/side/" which is green.

3. CDNI Metadata Bootstrapping

It is assumed that a well-known hostname to which MI requests should be sent is configured through the CDNI bootstrap data. Bootstrap information is sent through the CDNI CI, as described in the CDNI requirements document [I-D.ietf-cdni-requirements]. The MI APIs described herein are intended to be serviced by the MI running on that host.

Domain and Agent configurations must exist prior to Metadata creation/retrieval. Domains and Agents MAY be created as a part of an off-line business negotiation process or as a part of the CDNI bootstrapping process. Domain and Agent API descriptions are included in Appendix A and Appendix B, respectively. When the Domain

and Agent APIs described are used, access to the APIs SHOULD be secured using SSL with client authentication as described in the Security Considerations section.

Two sets of Agent configurations are also REQUIRED:

- Upstream Agent Configuration: Agent credentials for all external agents who require access to the local CDN MI, e.g. for dCDNs to retrieve Metadata or for uCDNs to trigger Metadata.

- Downstream Agent Configuration: Agent credentials for the local CDN to use when accessing uCDN MIs for retrieving Metadata or triggering Metadata responses. Separate credentials may be required for each uCDN and Domain combination from which content redirections may originate.

4. CDNI Metadata Management

The Metadata creation, modification, retrieval and removal protocols are defined in the following sections. All use a simple HTTP-based approach. The protocol, in general, SHOULD be data format agnostic. The examples shown herein use an XML representation for MI requests/responses, however, other well-defined representations (e.g., JSON) are also acceptable. The examples shown illustrate functionality required to support the data model described in Section 2, however, any protocol which allows for the forced retrieval, invalidation, and removal of Metadata could also be acceptable.

Metadata creation/update is distinguished from retrieval by the HTTP method. Metadata creation/update MUST use the POST method. Metadata retrieval MUST use the GET method. Metadata MUST be removed if the value field is empty (i.e., updating the value to be an empty string MUST force removal of the entire Metadata entry and all associated Base Address entries).

A trigger API is also specified to initiate retrieval of Metadata. The uCDN may issue a trigger to the dCDN to force (re)acquisition of Metadata by the dCDN. The trigger API MUST use the POST method.

In addition to being secured using SSL with client authentication as described in the Security Considerations section, the MI SHOULD also employ an additional Agent authentication mechanism to filter requests and results. In the examples shown below, HTTP basic authentication is used for Agent authentication, though other methods (e.g., HTTP digest authentication or URL hashing) could also be used.

4.1. Metadata API

The Metadata for a Domain is created/modified/retrieved using the "/CDNI/MI/metadata" API. The metadata API REQUIRES a single query string argument "domain" which specifies the name of the Domain to which the Metadata being created/modified/retrieved belongs. Three additional OPTIONAL arguments MAY also be provided when retrieving metadata: "name" which specifies the name of the Metadata field to create/modify/retrieve, "uri" which specifies the URI for which the Metadata must apply, and/or "agent" which specifies the agent(s) to which the Metadata is associated, as a comma separated list. The "agent" option MUST only be allowed for agents with full read/write permissions.

A simple XML representation of the information provided to the metadata creation/update API or returned from the metadata retrieval API is shown below:

```
<metadatas>
  <metadata>
    <uri></uri>
    <name></name>
    <values>
      <set>
        <value></value>
        <priority></priority>
      </set>
      ...
    </values>
    <must_enforce></must_enforce>
    <ttl></ttl>
    <agent></agent>
    <baseaddrs>
      <baseaddr></baseaddr>
      ...
    </baseaddrs>
  </metadata>
  ...
</metadatas>
```

Metadata retrieval for a Domain may be triggered using the "/CDNI/MI/trigger" API. The trigger API provides the information required to issue a metadata API retrieval request (i.e., the "domain", "name, and "uri" query string arguments). The metadata API REQUIRES a single query string argument "action" which specifies what type of action is being triggered.

The following actions MUST be supported:

- refresh: The dCDN MUST retrieve and update all Metadata specified in the trigger.

The following actions are considered OPTIONAL:

- preposition: The dCDN SHOULD retrieve and update all Metadata specified in the trigger.

A simple XML representation of the information provided to the trigger API is shown below:

```
<triggers>
  <trigger>
    <host></host>
    <domain></domain>
    <name></name>
    <uri></uri>
  </trigger>
  ...
</triggers>
```

4.1.1. Metadata Creation

The following example creates three new Metadata "color" for the "dcdn" Agent in the "acme" Domain, issued by the "ucdn" Agent to the uCDN MI:

```
POST /CDNI/MI/metadata?domain=acme HTTP/1.1
Host: ucdn.mi.cdni.example.com
Accept: */*
Authorization: Basic dWNkbjp4eHg=
Content-Length: 1053
Content-Type: application/x-www-form-urlencoded

<metadatas>
  <metadata>
    <uri>/grass/*</uri>
    <name>color</name>
    <values>
      <set>
        <value>brown</value>
        <priority>0</priority>
      </set>
    </values>
    <must_enforce>false</must_enforce>
    <ttl></ttl>
    <agent>dcdn</agent>
    <baseaddrs>
```

```
            <baseaddr>*.acme.com</baseaddr>
          </baseaddrs>
        </metadata>
        <metadata>
          <uri>/grass/on/the/other/side/*</uri>
          <name>color</name>
          <values>
            <set>
              <value>green</value>
              <priority>0</priority>
            </set>
          </values>
          <must_enforce>true</must_enforce>
          <ttl></ttl>
          <agent>dcdn</agent>
          <baseaddrs>
            <baseaddr>*.acme.com</baseaddr>
          </baseaddrs>
        </metadata>
        <metadata>
          <uri>/glasses/*</uri>
          <name>color</name>
          <values>
            <set>
              <value>violet</value>
              <priority>0</priority>
            </set>
          </values>
          <must_enforce>false</must_enforce>
          <ttl></ttl>
          <agent>ucdn</agent>
          <baseaddrs>
            <baseaddr>*.acme.com</baseaddr>
          </baseaddrs>
        </metadata>
      </metadatas>
```

4.1.2. Metadata Update

The following example updates the "color" Metadata for the "/glasses/*" portion of the "acme" Domain and "dcdn" Agent, issued by the "ucdn" Agent to the uCDN MI:

```
POST /CDNI/MI/metadata?domain=acme HTTP/1.1
Host: ucdn.mi.cdni.example.com
Accept: */*
Authorization: Basic dWNkbjp4eHg=
Content-Length: 361
Content-Type: application/x-www-form-urlencoded

<metadatas>
  <metadata>
    <uri>/glasses/*</uri>
    <name>color</name>
    <values>
      <set>
        <value>rose</value>
        <priority>0</priority>
      </set>
      <set>
        <value>violet</value>
        <priority>2</priority>
      </set>
    </values>
    <must_enforce>true</must_enforce>
    <ttl></ttl>
    <agent>ucdn</agent>
    <baseaddrs>
      <baseaddr>*.acme.com</baseaddr>
    </baseaddrs>
  </metadata>
</metadatas>
```

4.1.3. Metadata Refresh Trigger

The following example triggers the refresh of all "color" Metadata for the "acme" Domain. The trigger is issued by the "ucdn" Agent to the dCDN MI and is intended to force the "dcdn" Agent to retrieve Metadata from the uCDN MI.

```
POST /CDNI/MI/trigger?action=refresh HTTP/1.1
Host: dcdn.mi.cdni.example.com
Accept: */*
Authorization: Basic dWNkbjp4eHg=
Content-Length: 155
Content-Type: application/x-www-form-urlencoded

<triggers>
  <trigger>
    <host>ucdn.mi.cdni.example.com</host>
    <domain>acme</domain>
    <name>color</name>
    <uri></uri>
  </trigger>
</triggers>
```

The following example triggers the refresh of all Metadata for the URI "/grass/on/this/side", in the "acme" Domain. The trigger is issued by the "ucdn" Agent to the dCDN MI and is intended to force the "dcdn" Agent to retrieve Metadata from the uCDN MI.

```
POST /CDNI/MI/trigger?action=refresh HTTP/1.1
Host: dcdn.mi.cdni.example.com
Accept: */*
Authorization: Basic dWNkbjp4eHg=
Content-Length: 169
Content-Type: application/x-www-form-urlencoded

<triggers>
  <trigger>
    <host>ucdn.mi.cdni.example.com</host>
    <domain>acme</domain>
    <name></name>
    <uri>/grass/on/this/side</uri>
  </trigger>
</triggers>
```

4.1.4. Metadata Retrieval

The following example retrieves all "color" Metadata for the "acme" Domain. The request was issued by the "dcdn" Agent to the uCDN MI, and the results are filtered for the "dcdn" Agent:

```
GET /CDNI/MI/metadata?domain=acme&name=color HTTP/1.1
Host: ucdn.mi.cdni.example.com
Accept: */*
Authorization: Basic ZGNkbjp5eXk=

HTTP/1.1 200 OK
Content-Length: 714
Connection: close
Content-Type: text/xml

<metadatas>
  <metadata>
    <uri>/grass/*</uri>
    <name>color</name>
    <values>
      <set>
        <value>brown</value>
        <priority>0</priority>
      </set>
    </values>
    <must_enforce>false</must_enforce>
    <ttl></ttl>
    <agent>dcdn</agent>
    <baseaddrs>
      <baseaddr>*.acme.com</baseaddr>
    </baseaddrs>
  </metadata>
  <metadata>
    <uri>/grass/on/the/other/side/*</uri>
    <name>color</name>
    <values>
      <set>
        <value>green</value>
        <priority>0</priority>
      </set>
    </values>
    <must_enforce>true</must_enforce>
    <ttl></ttl>
    <agent>dcdn</agent>
    <baseaddrs>
      <baseaddr>*.acme.com</baseaddr>
    </baseaddrs>
  </metadata>
</metadatas>
```

The following example retrieves the Metadata for the URI "/grass/on/this/side" in the "acme" Domain. The request was issued by and the results are filtered for the "dcdn" Agent:

```
GET /CDNI/MI/metadata?domain=acme&uri=/grass/on/this/side HTTP/1.1
Host: ucdn.mi.cdni.example.com
Accept: */*
Authorization: Basic ZGNkbjp5eXk=

HTTP/1.1 200 OK
Content-Length: 361
Connection: close
Content-Type: text/xml

<metadatas>
  <metadata>
    <uri>/grass/*</uri>
    <name>color</name>
    <values>
      <set>
        <value>brown</value>
        <priority>0</priority>
      </set>
    </values>
    <must_enforce>false</must_enforce>
    <ttl></ttl>
    <agent>dcdn</agent>
    <baseaddrs>
      <baseaddr>*.acme.com</baseaddr>
    </baseaddrs>
  </metadata>
</metadatas>
```

4.1.5. Metadata Removal

The following example removes the violet "color" Metadata value for the URI "/glasses/*" and the "ucdn" Agent in the "acme" Domain by setting the value to an empty string, issued by the "ucdn" Agent to the uCDN MI:

```
POST /CDNI/MI/metadata?domain=acme HTTP/1.1
Host: ucdn.mi.cdni.example.com
Accept: */*
Authorization: Basic dWNkbjp4eHg=
Content-Length: 225
Content-Type: application/x-www-form-urlencoded

<metadatas>
  <metadata>
    <uri>/glasses/*</uri>
    <name>color</name>
    <values>
      <set>
        <value/>
        <priority>2</priority>
      </set>
    </values>
    <agent>ucdn</agent>
  </metadata>
</metadatas>
```

4.1.6. Metadata Errors

For any update, retrieval, or trigger request with malformed XML, the MI SHOULD respond with a 400 Bad Request status code. Ancillary unknown tags MAY be ignored.

For any trigger requests with an unsupported action, the MI SHOULD respond with a 403 Forbidden status code.

For any update or retrieval request for a uri/name/domain_id tuple which does not exist, the MI SHOULD respond with a 404 Not Found status code.

For any request which lacks a valid Agent authorization, the MI MUST respond with a 401 Unauthorized status code. This includes Agents with valid credentials, but who are marked as read_only and have requested Metadata associated with an alternate Agent through the specification of an "agent" query string parameter.

For any request which results in Metadata with an expired TTL, and for which an update cannot be retrieved from an upstream MI, the MI MUST respond to with a 500 Internal Server status code.

4.1.7. Metadata Prepositioning

The metadata creation/modification/removal APIs discussed above SHOULD only be used by uCDNs to manage Metadata in the local CDN.

Though the metadata creation/modification/removal APIs could be used to preposition metadata in dCDNs, the trigger API allows the uCDN to force refresh of the dCDN Metadata without directly posting Metadata to the dCDN. This allows the dCDNs to manage retrieval of Metadata using lazy updates.

dCDNs SHOULD NOT modify metadata dictated by a uCDN. dCDNs SHOULD only be assigned Agents with read_only access and SHOULD NOT have access to uCDN Domain or Agent APIs (restricted through the use of different SSL client authentication certificates, as described in the Security Considerations section).

5. Metadata Definitions

This section defines a base set of Metadata which SHOULD be supported by all CDNI implementations.

5.1. Origin Server

Content which is not pre-positioned must be acquired by the CDN from an origin server. The origin server Metadata specifies the base URL to which the content request URI may be appended in order to acquire the content. The origin server Metadata is defined as having the name "origin_server", with valid values containing a comma separated list of base URLs, and the must_enforce flag set to false:

```
name: origin_server
value: <url>
must_enforce: false
```

In some cases, multiple non-load balanced origin servers may be available for content acquisition. The origin server Metadata SHOULD support an unprioritized comma separate list of base URL values.

Note: The origin list Metadata is not a must_enforce, since, if the content cannot be acquired, there is no threat of unauthorized content distribution. Other Metadata or content pre-positioning may negate the need for origin server Metadata.

5.2. Activation Time

Content may be pre-positioned in anticipation of demand, however, the content license may have restrictions on delivery timeframe. The activation time Metadata specifies the first time at which the content may be delivered. The activation time Metadata is defined as having the name "activation_time", with valid timestamp values that MUST conform to RFC3339 [RFC3339], and the must_enforce flag set to

true:

```
name: activation_time
value: <timestamp>
must_enforce: true
```

If the activation time Metadata is set and the current time is less than the specified activation time, the CDN MUST respond to requests for that content with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

5.3. Deactivation Time

Content may be pre-positioned in anticipation of demand, however, the content license may have restrictions on delivery timeframe. The deactivation time Metadata specifies the last time at which the content may be delivered. The deactivation time Metadata is defined as having the name "deactivation_time", with valid timestamp values that MUST conform to RFC3339 [RFC3339], and the must_enforce flag set to true:

```
name: deactivation_time
value: <timestamp>
must_enforce: true
```

If the deactivation time Metadata is set and the current time is greater than the specified activation time, the CDN MUST respond to requests for that content with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

5.4. Administrative Disable

It is sometimes necessary to temporarily disable the distribution of certain media (e.g., inappropriate content, irregular access patterns, etc.) within a set accessibility period (i.e., the activation/deactivation time range). The administrative disable Metadata instructs the CDN not to deliver the specified content under any circumstances. The administrative disable Metadata is defined as having the name "admin_disable", with two valid values "true" and "false", and the must_enforce flag set to true:

```
name: admin_disable
value: [true | false]
must_enforce: true
```

If the administrative disable Metadata is set to "true", the CDN MUST respond to requests for that content with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

5.5. Delegation Depth

CSPs may wish to prevent cascading CDNs to enforce licensing restrictions. The delegation depth Metadata instructs the CDN to only delegate requests for the specified content if the delegation depth is greater than zero. If the depth is less than or equal to zero, a uCDN should not delegate requests for the specified content to any dCDNs under any circumstances. When distributing the delegation depth Metadata the uCDN MUST decrement the value of delegation depth by at least one if the current value is greater than zero. The uCDN MAY choose not to decrement the value if the value is already less than or equal to zero. The uCDN MAY decrement by more than one in order to get to zero. The delegation depth Metadata is defined as having the name "delegate_depth", with an integer value and the must_enforce flag set to true:

```
name: delegate_depth
value: <integer>
must_enforce: true
```

If the delegation depth Metadata is less than or equal to 0, the CDN MUST either service the content requests itself or respond to requests for that content with a 504 Server Busy status code (or equivalent for the given non-HTTP request protocol).

5.6. Footprint Filter

CSPs often purchase rights to content which are only valid when accessed from certain locations (e.g., within a given country or through a given access network). The footprint filter Metadata provides a list of valid source IP subnets from which content requests may be accepted. The footprint filter Metadata is defined as having the name "footprint", with valid values containing a comma separated list of IP subnet definitions, and the must_enforce flag set to true:

```
name: footprint
value: <ip_subnet> [, <ip_subnet>]...
must_enforce: true
```

If the footprint filter Metadata is set and the source address of a requesting client does not match any of the IP subnets listed, the CDN MUST respond to the content request with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

5.7. HTTP Header Filter

CSPs often desire the ability to filter requests based on the existence of specific HTTP header fields and values (e.g., User-Agent headers for device detection or custom headers inserted by client-side applications). The HTTP header filter Metadata provides a list of HTTP header names and values which MUST be verified. The HTTP header filter Metadata is defined as having the name "http_filter_headers", with valid values containing a comma separated list of HTTP header names and regular expression matching criteria definitions, and the must_enforce flag set to true:

```
name: http_filter_headers
value: <name>:<regex> [, <name>:<regex>]...
must_enforce: true
```

If the HTTP header filter Metadata is set and the HTTP headers of the content request do not match all of the filters specified, the CDN MUST respond to the content request with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

5.8. HTTP Header Logging

CSP client applications often include proprietary headers in their content requests (e.g., for user tracking or analaytics collection) which may be needed for business reasons (e.g., billing) or may be useful for debugging purposes. The HTTP header logging Metadata provides a list of HTTP header names whose values MUST be extracted and logged with the normal per-request information passed through the CDNI logging interface. The HTTP header logging Metadata is defined as having the name "http_logging_headers", with valid values containing a comma separated list of HTTP header names, and the must_enforce flag flag optionally set to true (depending on the application):

```
name: http_logging_headers
value: <name> [, <name>]...
must_enforce: [true | false]
```

If the HTTP header logging Metadata is set and the content request contains HTTP headers which match any of the header names listed, the CDN MUST extract all matching headers and add them to the per-request log message.

5.9. Protocol Filter

Though content is typically only accessible using specific a protocol (e.g., HTTP, RTMP, or RTSP), a CSP may wish to explicitly allow/

disallow access to certain content for a given protocol. The protocol filter Metadata provides a list of allowed protocols via which content may be delivered. The protocol filter Metadata is defined as having the name "protocol", with valid values containing a comma separate list of protocol strings, and the must_enforce flag set to true:

```
name: protocols
value: <protocol> [, <protocol>]...
must_enforce: true
```

If the protocol filter Metadata is set and the request protocol does not match any protocol in the list, the CDN MUST respond to the content request with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

5.10. SSL Required

CSPs which require delivery privacy may require dCDNs to support the same SSL configurations which were applied to the uCDN. The SSL required Metadata expresses the requirement to enforce SSL on content request connections and provides the necessary key and certificate information required for server authentication. The SSL required Metadata is defined as having the name "ssl_required", with valid values containing two URLs (comma separated) which point to the key and certificate, respectively, and the must_enforce flag set to true:

```
name: ssl_required
value: <key_url>,<cert_url>
must_enforce: true
```

If the SSL required Metadata is set and the request is not received over an SSL channel, the CDN MUST respond to the content request with a 403 Forbidden status code (or equivalent for the given non-HTTP request protocol).

Note: Retrieval of server key and certificate information SHOULD be performed in a secure manner. Retrieval could be implemented through the CDNI MI, however, this is not required.

5.11. SSL Client Authentication Required

CSPs which require client authentication may require dCDNs to support a SSL client authentication configuration which was applied to the uCDN. The SSL client authentication required Metadata expresses the requirement to enforce SSL client authentication on content requests and provides the necessary certificate authority (CA) information for authenticating clients. The SSL client authentication required

Metadata is defined as having the name "ssl_auth_required", with valid values containing a single URL which points to the CA certificate to be used in client verification, and the must_enforce flag set to true:

```
name: ssl_auth_required
value: <ca_url>
must_enforce: true
```

If the SSL client authentication required Metadata is set and the client certificate cannot be verified using the CA certificate, the CDN MUST respond with a handshake_failure alert.

5.12. URL Hash

TBD.

[Ed. Note: There are many proprietary URL hashing techniques in use today with varying timestamp formats, query string parameter names, hashing algorithm combinations, etc. A generic definition of URL hashing algorithm parameters, capable of supporting all algorithms would be best. An alternative of defining specific algorithms and assigning each and enumerated identifier would also work.]

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

There are a number of security concerns associated with the MI as Metadata may be used to influence CDNI request routing. Metadata may describe content acquisition parameters or content security restrictions. Altering Metadata or inhibiting Metadata discovery may impact content distribution. Some MI concerns include:

- intercepting and discarding Metadata requests to prevent content acquisition may be used as a denial of service attack,
- altering content acquisition Metadata to prevent content acquisition may be used as a denial of service attack, and
- spoofing content security Metadata to disable delivery restrictions may be used to circumvent rights management.

To combat these concerns, unauthorized access to the MI MUST be

prevented. The use of SSL with client authentication SHOULD be used for all MI APIs. Deployments in controlled environments where physical security and IP address white-listing is employed MAY choose not to use SSL. Different client authentication certificates SHOULD be used to protect access to Domain and Agent APIs, as well as uCDN access to the Metadata API, differently from dCDN access to the Metadata API. Deployments where uCDNs and dCDNs are mutually trusted entities (e.g., when uCDNs and dCDNs are controlled by the same corporate organization) MAY choose to use a single client authentication certificate.

8. Acknowledgements

The authors would like to thank Daniel Biagini, Susan He, Francois Le Faucheur, Kent Leung, Ben Niven-Jenkins, Gilles Bertrand, and Raj Nair for their helpful reviews and comments.

9. Appendix A: Domain API

Domain creation, modification, retrieval, and removal protocols are defined in the following sections. All use a simple HTTP-based approach. The protocol, in general, SHOULD be data format agnostic. The examples shown herein use an XML representation for MI requests/ responses, however, other well-defined representations (e.g., JSON) are also acceptable. The examples shown illustrate the functionality required to support the data model described in Section 2, however, any protocol which allows for the creation, modification, retrieval, and removal of Domains could also be acceptable.

Domain creation/update is distinguished from domain retrieval and removal by the HTTP method. Domain creation/update MUST use the POST method. Domain retrieval MUST use the GET method. Domain removal MUST use the DELETE method.

All Agents and Metadata MUST be associated with a Domain. A Domain is created/modified/retrieved/removed using the "/CDNI/MI/domain" API. The domain API REQUIRES a single query string argument "domain" which specifies the name of the Domain to be created/modified/ retrieved.

A simple XML representation of the information provided to the domain creation/update API or returned from the domain retrieval API is shown below:

```
      <domain>
        <provider></provider>
        <description></description>
      </domain>
```

9.1. Domain Creation

The following example creates a new Domain "acme":

```
      POST /CDNI/MI/domain?domain=acme HTTP/1.1
      Host: host.mi.cdni.example.com
      Accept: */*
      Content-Length: 81
      Content-Type: application/x-www-form-urlencoded

      <domain>
        <provider>acme</provider>
        <description>acme</description>
      </domain>
```

9.2. Domain Update

The following example updates the "acme" Domain:

```
     POST /CDNI/MI/domain?domain=acme HTTP/1.1
     Host: host.mi.cdni.example.com
     Accept: */*
     Content-Length: 209
     Content-Type: application/x-www-form-urlencoded

     <domain>
       <provider>acme rocket-powered products, inc</provider>
       <description>fine purveyors of high quality anvils, rubber bands,
                    bird seed, and rocket-powered footwear.</description>
     </domain>
```

9.3. Domain Retrieval

The following example retrieves the updated "acme" Domain information:

```
GET /CDNI/MI/domain?domain=acme HTTP/1.1
Host: host.mi.cdni.example.com
Accept: */*

HTTP/1.1 200 OK
Content-Length: 209
Connection: close
Content-Type: text/xml

<domain>
  <provider>acme rocket-powered products, inc</provider>
  <description>fine purveyors of high quality anvils, rubber bands,
               bird seed, and rocket powered footwear</description>
</domain>
```

The MI MAY support bulk retrieval of Domains through the use of a comma separated list of Domain names in the domain query string parameter.

9.4. Domain Removal

The following example removes the "acme" Domain:

```
DELETE /CDNI/MI/domain?domain=acme HTTP/1.1
Host: host.mi.cdni.example.com
Accept: */*
```

9.5. Domain Errors

Any update or retrieval request with malformed XML SHOULD respond with a 400 Bad Request status code. Ancillary unknown tags MAY be ignored.

Any update or retrieval request for a Domain which does not exist SHOULD respond with a 404 Not Found status code.

10. Appendix B: Agent API

Agent creation, modification, retrieval, and removal protocols are defined in the following sections. All use a simple HTTP-based approach. The protocol, in general, SHOULD be data format agnostic. The examples shown herein use an XML representation for MI requests/responses, however, other well-defined representations (e.g., JSON) are also acceptable. The examples shown illustrate the functionality required to support the data model described in Section 2, however, any protocol which allows for the creation, modification, retrieval,

and removal of Agents could also be acceptable.

Agent creation/update is distinguished from Agent retrieval and removal by the HTTP method. Agent creation/update MUST use the POST method. Agent retrieval MUST use the GET method. Agent removal MUST use the DELETE method and specify the Agent name(s) in the query string.

All Metadata MUST be associated with an Agent. An Agent is created/modified/retrieved/removed using the "/CDNI/MI/agent" API. The agent API REQUIRES a single query string argument "domain" which specifies the name of the Domain to which the Agent has access. In the case of DELETEs, the agent API also REQUIRES a query string argument "agent" which specifies the name(s) of the Agent(s) to remove, as a comma separated list.

A simple XML representation of the information provided to the agent creation/update API or returned from the agent retrieval API is shown below:

```
<agents>
  <agent>
    <username></username>
    <password></password>
    <read_only></read_only>
  </agent>
  ...
</agents>
```

10.1. Agent Creation

The following example creates three new Agents "ucdn", "dcdn1", and "dcdn2" for the "acme" Domain:

```
POST /CDNI/MI/agent?domain=acme HTTP/1.1
Host: host.mi.cdni.example.com
Accept: */*
Content-Length: 362
Content-Type: application/x-www-form-urlencoded

<agents>
  <agent>
    <username>ucdn</username>
    <password>xxx</password>
    <read_only>false</read_only>
  </agent>
  <agent>
    <username>dcdn1</username>
    <password>aaa</password>
    <read_only>false</read_only>
  </agent>
  <agent>
    <username>dcdn2</username>
    <password>bbb</password>
    <read_only>false</read_only>
  </agent>
</agents>
```

10.2. Agent Update

The following example updates the "dcdn1" and "dcdn2" Agents in the "acme" Domain:

```
POST /CDNI/MI/agent?domain=acme HTTP/1.1
Host: host.mi.cdni.example.com
Accept: */*
Content-Length: 245
Content-Type: application/x-www-form-urlencoded

<agents>
  <agent>
    <username>dcdn1</username>
    <password>yyy</password>
    <read_only>true</read_only>
  </agent>
  <agent>
    <username>dcdn2</username>
    <password>zzz</password>
    <read_only>true</read_only>
  </agent>
</agents>
```

10.3. Agent Retrieval

The following example retrieves the updated Agent information for the "acme" Domain:

```
GET /CDNI/MI/agent?domain=acme HTTP/1.1
Host: host.mi.cdni.example.com
Accept: */*

HTTP/1.1 200 OK
Content-Length: 360
Connection: close
Content-Type: text/xml

<agents>
  <agent>
    <username>ucdn</username>
    <password>xxx</password>
    <read_only>false</read_only>
  </agent>
  <agent>
    <username>dcdn1</username>
    <password>yyy</password>
    <read_only>true</read_only>
  </agent>
  <agent>
    <username>dcdn2</username>
    <password>zzz</password>
    <read_only>true</read_only>
  </agent>
</agents>
```

10.4. Agent Removal

The following example removes the "dcdn1" Agent from the "acme" Domain:

```
DELETE /CDNI/MI/agent?domain=acme&agent=dcdn1 HTTP/1.1
Host: host.mi.cdni.example.com
Accept: */*
```

10.5. Agent Errors

Any update or retrieval request with malformed XML SHOULD respond with a 400 Bad Request status code. Ancillary unknown tags MAY be ignored.

Any update or retrieval requests for an Agent which does not exist SHOULD respond with a 404 Not Found status code.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

11.2. Informative References

[I-D.davie-cdni-framework] Davie, B., Ed. and L. Peterson, Ed., "Framework for CDN Interconnection draft-davie-cdni-framework-01", October 2011.

[I-D.ietf-cdni-requirements] Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements draft-ietf-cdni-requirements-02", December 2011.

[I-D.ietf-cdni-use-cases] Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection draft-ietf-cdni-use-cases-04", March 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

Internet Engineering Task Force
Internet Draft
Intended status: Informational
Expires: April 2012

S. Manning
Huawei
R. Streijl
V. Prasad
P. Tarapore
AT&T
M. Geller
Cisco
R. Krishnan
Brocade

October 21, 2011

Additional Content Distribution Network Interconnection (CDNI) Requirements Based on ATIS CSF
draft-manning-cdni-additional-csf-reqs-00

Abstract

The purpose of Content Delivery Networks (CDNs) is to deliver content to end users in an efficient manner from the perspective of the network providers and with consistently good performance from the perspective of the end user. Due to footprint limitations of a single network provider, it may be necessary to interconnect CDNs between different providers. The Content Distribution Network Interconnection (CDNI) working group has been chartered to develop an interoperable and scalable solution for such CDN interconnection.

The requirements for CDN interconnection are being discussed and developed in various industry forums. One example is the Alliance for Telecommunications Industry Solutions (ATIS) Cloud Services Forum (CSF) which is looking at CDN interconnection requirements from the perspective of telecom providers. This document introduces some additional requirements to be included in the CDNI working group based on conclusions reached by ATIS CSF. The goal is for specifications developed by CDNI to successfully support some of the needs expressed by ATIS CSF as interpreted by the authors of this document.

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), its areas, and its working groups. Note that

other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on April 21, 2012.

Copyright Notice

Table of Contents

1. Introduction

[I-D.ietf-cdni-use-cases] and [I-D.ietf-cdni-requirements] describe the majority of use cases and requirements needed for the development of CDN interconnection specifications. This document introduces some additional requirements based on the conclusions reached by the Alliance for Telecommunications Industry Solutions (ATIS) Cloud Services Forum (CSF) which is looking at CDN interconnection requirements from the perspective of telecom providers. Based on the ATIS specification [CSF] as well as analysis and discussions within CSF, the authors have captured requirements suitable for the CDNI working group in this document.

Although working on largely the same problem area, ATIS CSF has a wider scope than the CDNI working group and considers interactions between Content Service Providers (CSPs) and CDNs as well as examining interface "domains" that include Operations & Customer Care (establishment of SLAs), Back Office (provisioning and charging), and aspects of the "data plane" [2]. Therefore, some of the ATIS CSF use cases and requirements fall outside the solution scope of the CDNI working group. However, some ATIS CSF requirements, especially in the CDNI logging and control interfaces, are well matched to the CDNI solution scope. The hope is that the inclusion of the appropriate requirements in this document will allow the CDNI working group specifications to support the ATIS community and help foster aligned solutions to the common CDNI problem. This will benefit the CDN community and encourage wider adoption of IETF CDNI standards.

2. Requirements Language

The key words "High Priority", "Medium Priority" and "Low Priority" in this document are to be interpreted in the following way:

- "High Priority" indicates requirements that are to be supported by the CDNI interfaces. A requirement is stated as "High Priority" when it is established by the working group that it can be met without compromising the targeted schedule for WG deliverables, or when it is established that specifying a solution without meeting this requirement would not make sense and would justify re-adjusting the WG schedule, or both. This is tagged as "[HIGH]".

- "Medium Priority" indicates requirements that are to be supported by the CDNI interfaces unless the WG realizes at a later stage that attempting to meet this requirement would compromise the overall WG schedule (for example it would involve complexities that would result in significantly delaying the deliverables). This is tagged as "[MED]".

- "Low Priority" indicates requirements that are to be supported by the CDNI interfaces provided that dedicating WG resources to this work does not prevent addressing "High Priority" and "Medium Priority" requirements and that attempting to meet this requirement would not compromise the overall WG schedule. This is tagged as "[LOW]".

3. Logging Interface Requirements

3.1. Logging Failures

One of ATIS CSF explicit requirements is for the logging interface to record individual actions on content items which includes unsuccessful deliveries. See [CSF] section 7.1.2, requirement R1. This leads to the following additional CDNI requirements:

LOG-A1 [HIGH] The CDNI Logging interface shall support logging of incomplete deliveries to User Agents performed by the Downstream CDN as a result of request redirection by the Upstream CDN.

LOG-A2 [MED] In the case of cascaded CDNs, the CDNI Logging interface shall support the Downstream CDN for reporting to the Upstream CDN logging for incomplete deliveries performed by the Downstream CDN itself as well as logging for incomplete deliveries performed by cascaded CDNs on behalf of the Downstream CDN.

3.2. Storage Resources

ATIS CSF requirements for the logging interface include the use of storage resources in the dCDN when such resources are requested by the uCDN. This is primarily useful for pre-positioning content. See [CSF] section 7.1.4, requirement R4 and R11. This leads to the following additional CDNI requirements:

LOG-A3 [MED] The CDNI Logging interface shall support logging of storage resources to the upstream CDN for deliveries where content is stored by the downstream CDN for delivery to User Agents. The information logged may include the type of storage (e.g., Origin, Intermediate, Edge, Cache) as well as the amount of storage (e.g., total GB, GB used, per time period, per content domain) all of which may impact the cost of the services.

LOG-A4 [MED] In the case of cascaded CDNs, the CDNI Logging interface shall support the Downstream CDN to report storage resources to the Upstream CDN where content is stored by the Downstream CDN itself as well as logging for storage resources when content storage is performed by cascaded CDNs on behalf of the Downstream CDN.

LOG-A5 [MED] The CDNI Logging interface shall support the upstream CDN to request the downstream CDN to return information on storage resources to the upstream CDN for deliveries where content is currently being stored by the downstream CDN for delivery to User Agents.

3.3. Performance Information

ATIS CSF requirements for the logging interface includes the reporting of performance statistics between the CDNs. This is especially important for monitoring common data traffic such as HTTP streaming sessions. See [CSF] section 7.1.2, requirement R6. This leads to the following additional CDNI requirement:

LOG-A6 [MED] The CDNI Logging interface shall support logging of performance data for deliveries to User Agents performed by the Downstream CDN as a result of request redirection by the Upstream CDN. Performance data may include various traffic statistics (the specific parameters are to be determined). The Logging interface shall support the upstream CDN to indicate the nature and contents of the performance data to be reported by the downstream CDN.

3.4. Delete Requests

ATIS CSF requirements for the logging interface includes recording explicit deletions of content (e.g., over the control interface). This leads to the following additional CDNI requirement:

LOG-A7 [HIGH] The CDNI Logging interface shall support logging of deleted objects from the downstream CDN to the upstream

CDN as a result of explicit delete requests on via the Control interface from the upstream CDN.

3.5. Extensible Information Fields

ATIS CSF requirements for the logging interface involves extensibility in the protocol to support implementation dependent information. See [CSF] section 7.1.2, requirement R2. This leads to the following additional CDNI requirements:

LOG-A8 [HIGH] The CDNI Logging interface shall support extensibility to allow proprietary information fields to be carried. These information fields must be agreed upon ahead of time between the corresponding CDNs.

LOG-A9 [HIGH] The CDNI Logging interface shall support the exchange of extensible log file formats to support proprietary information fields. These information fields must be agreed upon ahead of time between the corresponding CDNs.

4. Control Interface Requirements

4.1. Deletion of Objects

The uCDN may explicitly command the dCDN to delete certain content objects. ATIS CSF views that the deletion of objects is particular sensitive to CDN providers and the interface operation needs some clarifications. For example, it might take some finite amount of time to process deletions in the dCDN. During this time, the uCDN may assume that the dCDN will continue to deliver the content marked for deletion. But once the delete acknowledgement is received, the uCDN should be certain that no more deliveries will take place out of the dCDN and all copies of the content have been completely removed. The following CDNI requirement makes these assumptions clear:

CNTL-A1 [HIGH] The CDNI Control interface shall support the process by which the uCDN receives confirmation that the deletion of all copies of content have been done by the dCDN upon request by the uCDN. The confirmation receipt should be supported through a synchronous and/or asynchronous mechanism and should include a success or failure indication. The failure indication is used if the dCDN cannot delete the content.

Another situation is where an object is made up of a collection of sub-objects. The dCDN may fail to delete the entire object. In this case, a partial delete indication should be sent to the uCDN specifying which sub-objects were successfully deleted. The following CDNI requirement makes this clear:

CNTL-A2 [MED] The CDNI Control interface should support the Downstream CDN to indicate to the Upstream CDN a list of sub-objects that were successfully deleted and a list of sub-objects that were unsuccessfully deleted in the case of an object made up of a collection of sub-objects was not fully deleted by the Downstream CDN.

Finally, there is the case where an uCDN wishes to purge all content associated with a particular dCDN without issuing multiple delete requests for each and every content object. The CDN pair, however, continues to have a business relationship and therefore may elect to maintain the established CDNI session. The following CDNI requirement supports this:

CNTL-A3 [MED] The CDNI Control interface should support the Upstream CDN to efficiently request that the Downstream CDN that all content stored in the Downstream CDN on behalf of the Upstream CDN be deleted without enumeration of each individual object. Further, a single delete request may operate across many objects based on parameters such as content type, content provider name, content domain, etc.

4.2. Reservation of Resources

ATIS CSF has requirements for the uCDN to ask the dCDN to reserve bandwidth/storage resources in anticipation of content deliveries. For example, this may be important for the delivery of live streaming content. This is seen in [CSF] section 7.2.3, requirement R12. The following CDNI requirement supports this:

CNTL-A4 [MED] The CDNI Control interface shall support the Upstream CDN to request that the Downstream CDN to reserve capacity at some future time in terms of streaming bandwidth between the CDNs and/or storage resources in the downstream CDN prior to content delivery.

5. Security Considerations

This document adds no additional security considerations beyond those found in [I-D.ietf-cdni-use-cases] and [I-D.ietf-cdni-requirements].

6. IANA Considerations

This document makes no request of IANA.

7. References

7.1. Normative References

[I-D.ietf-cdni-requirements]
K. Leung, K. and Lee Y. (Editors), "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-00 (work in progress), September 2011.

[I-D.ietf-cdni-use-cases]
Bertrand, G. (Editor), Stephan, E., Watson, G., Burbridge, T., Eardley, P., and Ma, K., "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

7.2. Informative References

[CSF] Tarapore, P. and Munson G. (Editors), "CDN Interconnection Use Case Specification and High Level Requirements", Alliance for Telecommunications Industry Solutions: ATIS-0200003, June 2011.

8. Acknowledgments

The authors wish to thank Gary Munson, Andrew White, Spencer Dawkins, and Jincheng Li, as well as the members of the ATIS Cloud Services Forum for their discussions and input.

Authors' Addresses

Serge Manning
Huawei US Research Center
Plano, TX

Email: sergem913@gmail.com

Robert Streijl
AT&T
Chicago, IL

Email: rs0608@att.com

Vishwa Prasad
AT&T
Middletown, NJ

Email: vp2613@att.com

Percy Tarapore
AT&T
Middletown, NJ

Email: pt5947@att.com

Mike Geller
Cisco Systems

Email: mgeller@cisco.com

Ramki Krishnan
Brocade Communications
San Jose, CA

Email: ramk@brocade.com

CDNI WG
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2012

S. Previdi, Ed.
F. Le Faucheur
Cisco Systems, Inc.
A. Guillou
SFR
J. Medved
Juniper Networks, Inc.
October 24, 2011

CDNI Footprint Advertisement
draft-previdi-cdni-footprint-advertisement-00

Abstract

This document describes the use of BGP for Content Delivery Networks (CDNs) in order to advertise information about footprint and connectivity to footprint in the context of CDNI.

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 April 26, 2012.

Copyright Notice

Table of Contents

1. Introduction

The IETF CDN Interconnection (CDNI) Working Group is chartered to develop specifications for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users.

[I-D.jenkins-cdni-problem-statement] outlines the problem area that the CDNI working group is chartered to address. [I-D.bertrand-cdni-use-cases] discusses the use cases for CDN Interconnection and [I-D.davie-cdni-framework] discusses the technology framework for the CDNI solution and interfaces.

When an upstream CDN (uCDN) receives a request from a user, it has to determine what is the downstream CDN (dCDN) to which the request is to be redirected. This CDN selection decision can take into account various criteria such as administrative preferences (for example based on the commercial arrangements between the uCDN and candidate dCDNs including associated request handling costs) and/or such as whether candidate CDNs are have caches that are topologically close to the user and capable of handling that request. Therefore, as discussed in section "Dynamic Footprint Discovery" of [I-D.davie-cdni-framework], there are situations where being able to dynamically discover the set of requests that a given dCDN is willing and able to serve is beneficial. As also discussed in [I-D.davie-cdni-framework], this information could be potentially provided by the dCDN in response to a query by the uCDN, or the information (or its changes) could be spontaneously advertised by the dCDN.

The proposal outlined in this document makes use of Multiprotocol-BGP (MP-BGP [RFC4760]) in order for CDNs and/or ISPs to advertise their footprint information as well as for CDNs to advertise their connectivity to these footprints. In addition CDNs use MP-BGP advertisements to represent their interconnectivity.

2. CDNI Mesh and MP-BGP

CDNI enables CDNs to communicate in order to deliver content in a collaborative mode. In this document, we refer to a CDNI Mesh as the set of CDNs participating into CDNI and using MP-BGP sessions between them in accordance with the approach defined in this document. A CDNI Mesh has no requirements in terms of topology, i.e.: the mesh can be partial, full or hierarchical.

CDNI Mesh will make use of Multiprotocol-BGP (MP-BGP [RFC4760]) for

the exchange of footprint and connectivity information.

We define a new Address Family (CDNI-AF, TBD) and a new NLRI that will carry either CDN Footprint or CDN Connectivity advertisements. The NLRI will have a NLRI type (i.e.: CDNI-footprint and CDNI-connectivity) so to distinguish footprint and connectivity advertisements.

The advantage of using a separate address family is to isolate CDNI information from regular BGP-4 Internet information so to not compromise in any way the security and reliability of the current BGP information exchange used for IP network layer routing.

The advantage of having separate footprint and connectivity information is that a CDN needs not originate (and update) footprint information each time there's a change in the way a CDN is connected to other CDNs. E.g.: if all footprint information was to be exchanged between CDNs, it would consist of a very large amount of prefixes advertised (and re-advertised) each time a CDNI interconnection changes in the CDNI Mesh.

When an existing CDN connection is removed or when a new connection between two CDNs is established, the only advertisements that need to be updated are the ones concerning the connectivity.

While the footprint information is expected to be relatively stable, the CDN Mesh (i.e.: the connectivity between CDNs) and the connectivity between the CDN and the footprints may be impacted by network events. Also, the connectivity between CDNs may be affected by the CDN selection policy which may be modified relatively frequently.

CDNI Connectivity advertisements allow the CDNI Mesh to scale by adapting easily to topology changes. In fact, just a few number of connectivity advertisements are used by each CDN which makes the CDNI MP-BGP scheme very scalable.

3. CDNI Information

The CDNI Information is of two types:

Footprint Information, stored in the CDNI (MP-BGP) Footprint Database.

Connectivity Information, stored in the CDNI (MP-BGP) Connectivity Information Database.

3.1. Footprint Information

Footprint Information (FI). The CDN Footprint Information refers to the set of prefixes (with all their BGP attributes) that the CDN is capable of, and willing to deliver content to in a given region and or in a given Autonomous System. Note that a CDN may be capable and willing to serve content to more than one footprint.

Example: if CDN-A delivers content to ISP-A users, then CDN-A footprint consists of all prefixes owned and connected to ISP-A.

Footprint Information is therefore inferred from the BGP-4 Internet database. It is assumed that the CDN will have a BGP-4 feed with Internet prefixes that are necessary in order for the CDNI Mesh to operate and from which it will be able to derive the different footprints.

The CDN will maintain a database with footprint information that is separate from the regular IP BGP database. The CDNI footprint information database uses the CDNI-AF MP-BGP address family.

In addition, a CDN may want to advertise to other CDNs part or all of its footprint information. For example, a CDN may want to give a better granularity of the prefixes of its footprint (e.g.: longer masks) or may want to add more attributes (e.g.: communities and extended communities) to its footprint information. Therefore the CDN is capable of originating Footprint Advertisements (from its CDNI Footprint Database) and send them to its neighbors of the CDNI-Mesh.

The CDNI Footprint Information Database includes footprint information inferred by the BGP-4 (Internet) database as well as footprint information explicitely advertised by neighboring CDNs.

3.2. Connectivity Information

Connectivity Information refers to how the CDN is connected to a footprint (and to which footprints). This information needs to be advertised by the CDN to the rest of the CDNI Mesh so that every CDN knows which CDN is connected to which footprints.

It has to be noted that by "connectivity" we do not intend physical direct connectivity between the CDN and the footprint but rather the ability to deliver content to the footprint. Connectivity Advertisements are sent through MP-BGP and using CDNI-AF advertisements.

Connectivity Information is stored in the CDNI Connectivity Database which contains the information originated by the CDN and the

information received from the other CDNs in the CDNI Mesh.

4. CDNI MP-BGP

This section describes the CDNI Footprint and Connectivity Information in the CDNI MP-BGP databases. Two databases are used:

CDNI Footprint Database that contains footprint that is either derived from BGP-4 Internet table or received from other CDN through CDNI Footprint Advertisements.

CDNI Connectivity Database that contains the advertisements made by each CDN describing how they are connected to footprints.

4.1. CDNI MP-BGP Footprint Information and Advertisements

Footprint information first comprises IP prefixes as known in the BGP-4 database and that need to be translated into the CDNI-AF format and stored in the CDNI Footprint database. BGP-4 information is inserted in the CDNI Footprint database and all BGP attributes of each original route are preserved (e.g.: AS_PATH, MED, Communities, Extended Communities). In addition, a CDN may add more attributes to the CDNI Footprint database routes.

As discussed earlier, a CDN may want to explicitly advertise footprint information to the CDNI-Mesh (as explained in Section 3.1). When it does so, this information is also incorporated by the in its CDNI Footprint database by a CDN receiving these advertisments. However, it is expected that a CDN acquires most of the footprint information from the BGP-4 Internet table. So we expect limited usage of footprint advertisements between CDNs.

For example, a CDN (or ISP) may originate MP-BGP footprint advertisement including a Community attribute representing the location of the prefixes or the type of user connectivity (e.g.: fiber vs. cable vs. dsl vs Mobile3G vs Mobile 4G). Alternatively, such information could be delivered initially by the ISP in the BGP-4 database.

4.1.1. CDNI Footprint Attributes: Footprint Identifier

Footprints are associated to Autonomous Systems. Therefore, the identifier of a footprint is its Autonomous System Number (ASN). When the CDN creates the CDNI Footprint database, it will assign to each prefix, a new (TBD) Extended Community carrying the Footprint Identifier.

Footprint Identifier is derived from the Autonomous System Number (ASN) of the original route. When inferring the CDNI footprint information from the regular BGP-4 Internet database, the footprint identifier is derived from the first ASN in the AS_PATH of the prefix.

The role of the Footprint Identifier is to group all prefixes part of the same footprint under a unique identifier. This allows a CDN to claim connectivity to the footprint by just specifying the FI rather than each individual prefix of the footprint.

Footprint Identifier may also be used in order to describe a finer granularity than the ASN. Example: a CDN or an ISP participating into the CDNI Mesh, may want to originate footprint advertisement with a Footprint Identifier describing a region of its footprint (e.g.: an ISP may have multiple peering points in different locations and may want to partition its footprint so to represent geographical groups.)

For that purpose multiple Footprint Identifiers are used (e.g.: a footprint representing Los Angeles area and another footprint representing New York City area). These two footprints MUST be understood as part of the same ISP but representing different groups of prefixes.

Using separate Footprint Identifiers (one for LA prefixes and one for NYC prefixes) allows the CDNI Mesh to handle the footprints separately even if they belong to the same ISP. Footprint Identifier MUST be unique across the CDNI Mesh and therefore are numbered using the ISP AS numbers followed by additional bit space allowing more footprint identifiers per ISP.

4.1.2. CDNI Footprint Attributes: Origin_AS_PATH

A new MP-BGP attribute (TBD) is defined and called Origin_AS_PATH. This attribute contains the prefix AS_PATH value that is present on the CDNI footprint database.

The Origin_AS_PATH is used when a CDN originates a CDNI Footprint Advertisement. The AS_PATH of the new advertisement follows the BGP rules (i.e.: it is created with the CDN ASN and further updated at each AS hop) while the Origin_AS_PATH contains the AS_PATH of the original prefix.

4.1.3. Multihomed Prefixes

In some cases, a given prefix may be part of different footprint if it represents a customer connected to two separate ISPs. In some

cases it is useful to preserve this information and allow both prefixes advertisements in the BGP database. However, due to BGP Path Selection rules, when a BGP speaker receives two or more advertisements for the same prefix, it selects one and ignore the others.

In order to prevent this to happen a Route Distinguisher may be used in the advertisement so that, from a BGP selection perspective, the prefix advertisements are not considered being equal.

4.2. CDNI MP-BGP Connectivity Information and Advertisements

Once footprint information is known in the CDNI Mesh, each CDN should advertise its connectivity to the footprints it has access to. The CDN maintains a MP-BGP CDNI Connectivity Database with entries describing its connectivity to footprints.

4.2.1. CDNI Connectivity Prefix

When a CDN wants to advertise its footprint connectivity it originates a MP-BGP advertisement containing a prefix and a set of attributes. The prefix it uses MUST be a prefix in the address space owned by the CDN. A CDN willing to advertise different set of footprints connectivity may use different prefix advertisements each with its set of attributes.

The Connectivity prefix(es) the CDN originates may contain any standard MP-BGP attribute and it MUST contain a newly defined attribute: Connected Footprints.

4.2.2. CDNI Connectivity Attribute: Connected Footprints

Connected Footprints attribute describes the set of Footprint Identifiers (FIs) the CDN claims connectivity to.

The Connected Footprint attribute (CF) is a set of Footprint Identifiers which means a set of Extended Communities as defined in Section 4.1.1.

The propagation of CDN Connectivity advertisements throughout the CDNI Mesh is done according to standard MP-BGP rules and the inter-CDN connectivity will be reflected in the MP-BGP attributes (e.g.: AS_PATH will describe the different CDNs the advertisement traversed during its propagation in the CDNI Mesh thus describing the inter-CDN connectivity).

4.2.3. CDNI Connectivity Advertisement Attributes: Origin_AS_PATH

The CDNI Connectivity Advertisement contains a new (TBD) attribute called Origin_AS_PATH that contains the AS_PATH value describing the distance (expressed in AS Hop Count) between the CDN and the advertised connected footprint.

This attribute will allow remote CDNs to understand how this CDN is distant (or close, in terms of AS hop count) to the footprint.

The regular AS_PATH attribute of the Connectivity Advertisement is updated during its propagation in the CDNI Mesh so to prevent BGP message loops (according to BGP rules).

5. CDNI Topology Example

The figure below gives an example how CDNs collaborate and how they create their CDNI Footprint and Connectivity databases.

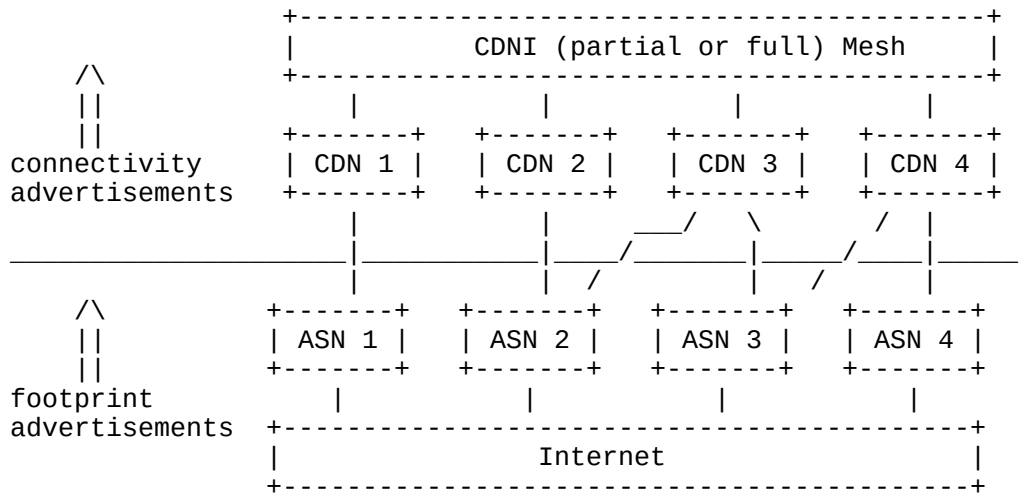


Figure 1: Footprint and Connectivity Advertisements

In the figure above 4 CDNs are connected to a set of 4 different footprints. Each CDN is capable of inferring the footprint information from the BGP-4 Internet table and will create a CDNI Footprint Database where a Footprint Identifier will be assigned to each prefix of the footprints. The Footprint Identifier to each prefix will be assigned based on the ASN of the prefix.

In addition, each CDN advertises its connectivity to the footprint.

All CDNI information (footprint and connectivity) is then known in the entire CDNI Mesh.

6. CDNI MP-BGP Operations

Connectivity advertisements and, when necessary, Footprint Advertisements consist of MP-BGP update messages CDNs advertise to the CDNI Mesh (following standard MP-BGP propagation rules).

Footprint consists of prefixes known in the CDNI Footprint Database (MP-BGP). CDNs advertises these prefixes and can use standard BGP attribute to attach more information to these prefixes. In addition a new Extended Community Type is defined so to convey the Footprint Identifier that associate each prefix to a given footprint.

Connectivity advertisements are originated using one or more prefixes the CDN will use in order to convey the description of its connectivity to a footprint. The prefix has the solely purpose to convey the connectivity information of the CDN (i.e.: the prefix itself is not to be used for routing or selection purposes). The footprint connectivity of a CDN is expressed in the Extended Community Attribute type "Connected Footprint" which consists of the set of Footprint Identifiers the CDN is connected/have access to. The Extended Community being additive, more than one Footprint Identifier is allowed in the CDN Connectivity advertisement.

Also, the Origin_AS_PATH attribute reflects how the CDN is effectively distance/close to the footprint from a network layer perspective.

MP-BGP sessions are established between CDNs. MP-BGP requires an Autonomous System (AS) number that is unique across all CDNs. Therefore each CDN participating into the CDNI Mesh MUST have a unique AS number.

6.1. Internal and External MP-BGP Sessions

CDNs establish external MP-BGP sessions with each others. The MP-BGP session has to be established using the BGP Capabilities specifying the speaker is capable of MP-BGP and for the CDNI Address Family. In each of the CDNs at least one MP-BGP speaker will be available ensuring connectivity to the CDNI Mesh.

Internal MP-BGP sessions can be used inside a CDN for propagating footprint and connectivity advertisements. Same mechanisms such as route reflectors and/or confederations can be used internally to a CDN.

A CDN that is operated by an ISP may use the same ASN than the one currently used by the ISP.

6.2. CDNI MP-BGP NLRI

Two types of CDNI NLRIs are defined: footprint and connectivity.

6.2.1. CDNI Footprint NLRI and Attributes

This CDNI NLRI Type (TBD) describes footprint information which consists of an IPv4 or IPv6 address prefix. CDNI Footprint Advertisement includes following information (note well that in this section we do not aim to describe in details the format of the NLRI bur rather focus on the kind of information it should contain):

RD:ipv4 or RD:ipv6 addresses
BGP attributes such as: AS_PATH, NEXT_HOP, MED,
Community, ExtCommunity.
Origin_AS_PATH
Footprint Identifiers

where:

RD is the Route Distinguisher of the route. When used, it allows to distinguish among multiple advertisements of the same prefix (for the multihomed case). The RD is present in CDNI NLRI using CDNI-AF with SAFI value set to 128.

Origin_AS_PATH contains the AS_PATH value that is present in the CDNI Footprint Database for this prefix.

Footprint Identifier describes to which footprint this prefix belongs to. The footprint Identifier represents the ASN of origin of the prefix.

Footprint NLRIs are propagated in the CDNI Mesh according to standard MP-BGP rules and MP-BGP attributes such as AS_PATH, NEXT_HOP, MED, Local Preference, etc, are updated and used according to the standard MP-BGP mechanisms. For example, AS_PATH is updated and checked so to avoid messaging loops.

6.2.2. CDNI Connectivity NLRI and Attributes

The CDNI Connectivity NLRI Type (TBD) describes how the CDN connects to a given footprint. Each CDN originates one or more prefixes whose purpose is to convey attributes describing how the CDN can reach the footprint. The Connectivity advertisment includes following information:

RD:ipv4 or RD:ipv6 addresses
BGP attributes such as: AS_PATH, NEXT_HOP, MED,
Community, ExtCommunity.
Origin_AS_PATH
Connected Footprints

where:

RD is the Route Distinguisher of the route. When used, it allows to distinguish among multiple advertisements of the same prefix (for the multihomed case). The RD is present in CDNI NLRI using CDNI-AF with SAFI value set to 128. It has to be noted that the CDNI Advertisements makes no use of the route distinguisher and therefore it has to be set to all zero.

BGP attributes convey information related to the connectivity of the CDN to the footprint.

Origin_AS_PATH contains the AS_PATH that describes how the CDN is connected to the footprint (i.e.: what is the AS hop count).

Connected Footprints describe which footprints the CDN has connectivity to.

7. Example of CDNI Mesh

In this section an example of a CDNI Mesh is described as well as the CDNI Information that will be originated.

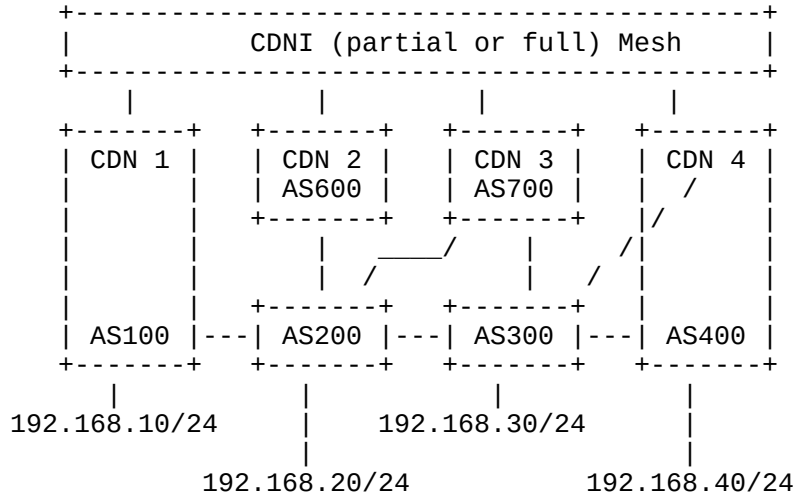


Figure 2: CDNI Mesh Example

We assume different Autonomous Systems (representing different ISPs) with their own prefix space. We also assume four CDNs that are, for some of them, on their own AS while others share the AS of their ISP (which describe the case where the CDN is managed by the ISP). The Internet is represented by the interconnectivity of the Autonomous System.

7.1. CDNI Footprint Information and Advertisements

Footprint information will be inferred by each CDN from the BGP-4 database. For example, the CDNI Footprint Database in CDN4, located in Autonomous System 400, is as follows:

```
              IPv4 prefix:                192.168.10/24
              AS_PATH:                    300 200 100
              CDNI Footprint Identifier:  100:000

              IPv4 prefix:                192.168.20/24
              AS_PATH:                    300 200
              CDNI Footprint Identifier:  200:000

              IPv4 prefix:                192.168.30/24
              AS_PATH:                    300
              CDNI Footprint Identifier:  300:000

              IPv4 prefix:                192.168.40/24
              AS_PATH:
              CDNI Footprint Identifier:  400:000
```

where:

IPv4 Prefix is the footprint prefix information.

AS_PATH is the existing BGP AS_PATH attribute (from the BGP-4 database) containing the set of ASNs the update has traversed in the Internet.

CDNI Footprint Identifier represents the footprint as a whole. All prefixes part of the same footprint will share the same Footprint Identifier. For CDNI purposes, CDN4 may want to advertise more information about the inferred footprint so to tell the CDNI Mesh more information about location of the footprint. For example, part of the address space 192.168.40/24 is located in NYC, part is located in Chicago and part in LA. CDN4 may originate new CDNI Footprint Information such as:

```
IPv4 prefix:                192.168.40.64/26
AS_PATH:
CDNI Footprint Identifier:  400:001

IPv4 prefix:                192.168.40.128/26
AS_PATH:
CDNI Footprint Identifier:  400:002

IPv4 prefix:                192.168.40.192/26
AS_PATH:
CDNI Footprint Identifier:  400:003
```

When CDN4 sends out the above advertisements, it will have to update both the AS_PATH attribute (in order to prevent BGP message loops as well as the Origin-AS_PATH attribute so to preserve the original AS_PATH. Therefore the advertisements CDN4 will send out will be as follows:

```
IPv4 prefix:                192.168.40.64/26
AS_PATH:                    400
CDNI Footprint Identifier:  400:001
Origin_AS_PATH              400

IPv4 prefix:                192.168.40.128/26
AS_PATH:                    400
CDNI Footprint Identifier:  400:002
Origin_AS_PATH              400

IPv4 prefix:                192.168.40.192/26
AS_PATH:                    400
CDNI Footprint Identifier:  400:002
Origin_AS_PATH              400
```

When the advertisement is received by the CDNI Mesh neighbors it will also be propagated. For example, CDN3 may receive these advertisements and send them to CDN2. At CDN2 the advertisements will contain following information:

```
IPv4 prefix:                192.168.40.64/26
AS_PATH:                    300 400
CDNI Footprint Identifier:  400:001
Origin_AS_PATH              400

IPv4 prefix:                192.168.40.128/26
AS_PATH:                    300, 400
CDNI Footprint Identifier:  400:002
Origin_AS_PATH              400

IPv4 prefix:                192.168.40.192/26
AS_PATH:                    300 400
CDNI Footprint Identifier:  400:002
Origin_AS_PATH              400
```

where the AS_PATH attributes reflect the path taken by the advertisement in the CDNI-Mesh while the Origin_AS_PATH reflects the AS_PATH the prefix had at its point of origin (ASN 400).

7.2. Connectivity Information and Advertisements

Connectivity advertisements have the dual purpose of describing the CDN connectivity to the footprint and the connectivity to other CDNs. For example, CDN4 connectivity advertisement will describe the footprint connectivity to Footprint Identifiers 400:001, 400:002 and 400:003. The following is the advertisement CDN4 will send to its MP-BGP neighbors:

```
IPv4 prefix:          192.168.4.4/32
AS_PATH:              400
Connected footprints: 400:001, 400:002, 400:003
```

where:

IPv4 Prefix is the IPv4 address identifying the CDN (or part of it) in the CDNI Mesh.

AS_PATH contains the AS numbers this update traversed (including the AS where this connectivity advertisement has been originated).

Connected Footprints contains the set of footprint identifiers this CDN is directly connected to.

*** Editor’s Note: a mechanism in order to express preference or costs to footprints is needed: either through a ranking sequence or through explicit preference cost or weight.

Alternatively, CDN4 may want to advertise 3 distinct Connectivity Advertisements for each of the footprints its connected to:

```
IPv4 prefix:          192.168.4.1/32
AS_PATH:              400
Connected footprints: 400:001

IPv4 prefix:          192.168.4.2/32
AS_PATH:              400
Connected footprints: 400:002

IPv4 prefix:          192.168.4.3/32
AS_PATH:              400
Connected footprints: 400:003
```

Each prefix will carry a different Footprint Identifier so that CDN4 can separate these advertisements.

Each CDN may also advertise a set of communities representing the capabilities of the CDN. It is possible for a transit CDN to manipulate the set of communities during its propagation. Example: a CDN, prior to propagate the connectivity advertisement of another CDN, may strip one or more capabilities from the original advertisement. This will allow to enforce a given path selection by upstream CDNs.

8. Compliance with CDNI Requirements

[I-D.ietf-cdni-requirements] outlines the requirements for the solution and interfaces to be specified by the CDNI working group. This section identifies the relevant requirements from that document and discusses compliance by the solution proposed in this document.

[Editor's Note: Text is to be added when requirements-03 is available. This needs to discuss the requirements labeled R27, R28, R29 and R30 as of requirements-02].

9. IANA Considerations

none

10. Security Considerations

This draft does not affect the BGP security model.

11. Acknowledgements

The authors would like to recognize Bruce Davie for his contributions.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

12.2. Informative References

[I-D.bertrand-cdni-use-cases] Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-bertrand-cdni-use-cases-02 (work in progress), July 2011.

[I-D.davie-cdni-framework] Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[I-D.ietf-cdni-requirements] Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-01 (work in progress), October 2011.

[I-D.jenkins-cdni-problem-statement] Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-jenkins-cdni-problem-statement-02 (work in progress), March 2011.

Authors' Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico 200
Rome 00191
IT

Email: sprevidi@cisco.com

Francois Le Faucheur
Cisco Systems, Inc.
Greenside, 400 Avenue de Roumanille
Sophia Antipolis 06410
FR

Email: flefauch@cisco.com

Allan Guillou
SFR
40-42 Quai du Point du Jour
Boulogne-Billancourt 92659
FR

Email: allan.guillou@sfr.com

Jan Medved
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: jmedved@juniper.net

CDNI WG
Internet-Draft
Intended status: Standards Track
Expires: March 23, 2013

S. Previdi, Ed.
F. Le Faucheur
J. Medved
Cisco Systems, Inc.
A. Guillou
SFR

September 19, 2012

CDNI Footprint Advertisement
draft-previdi-cdni-footprint-advertisement-02

Abstract

This document describes the use of BGP for Content Delivery Networks (CDNs) in order to advertise information about footprint and connectivity to footprint in the context of CDNI.

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 March 23, 2013.

Copyright Notice

Table of Contents

1. Introduction

The IETF CDN Interconnection (CDNI) Working Group is chartered to develop specifications for interconnecting standalone CDNs so that their collective CDN footprint can be leveraged for the end-to-end delivery of content from Content Service Providers (CSPs) to End Users.

[I-D.jenkins-cdni-problem-statement] outlines the problem area that the CDNI working group is chartered to address. [I-D.bertrand-cdni-use-cases] discusses the use cases for CDN Interconnection and [I-D.davie-cdni-framework] discusses the technology framework for the CDNI solution and interfaces.

When an upstream CDN (uCDN) receives a request from a user, it has to determine what is the downstream CDN (dCDN) to which the request is to be redirected. This CDN selection decision can take into account various criteria such as administrative preferences (for example based on the commercial arrangements between the uCDN and candidate dCDNs including associated request handling costs) and/or such as whether candidate CDNs are have caches that are topologically close to the user and capable of handling that request. Therefore, as discussed in section "Dynamic Footprint Discovery" of [I-D.davie-cdni-framework], there are situations where being able to dynamically discover the set of requests that a given dCDN is willing and able to serve is beneficial. As also discussed in [I-D.davie-cdni-framework], this information could be potentially provided by the dCDN in response to a query by the uCDN, or the information (or its changes) could be spontaneously advertised by the dCDN.

The proposal outlined in this document makes use of Multiprotocol-BGP (MP-BGP [RFC4760]) in order for CDNs and/or ISPs to advertise their footprint information as well as for CDNs to advertise their connectivity to these footprints. In addition CDNs use MP-BGP advertisements to represent their interconnectivity.

2. Terminology

The following terminology is used in this document:

CDNI Mesh: the set of CDNs participating into CDNI and using MP-BGP sessions between them in accordance with the approach defined in this document. A CDNI Mesh has no requirements in terms of topology, i.e.: the mesh can be partial, full or hierarchical.

Footprint (FP): The exhaustive set of Prefixes a CDN is willing to serve.

Footprint Element (FPE): Arbitrary set of prefixes with attributes. They can be implicit (inferred from BGP) or explicit (advertised).

Implicit Footprint Element Advertisement (FPE Implicit Advertisement): Footprint Element information derived from the BGP database.

Footprint Element Advertisement (FPE-Adv): MP-BGP Message used by a CDN in order to advertise or withdrawn Footprint Elements and their attributes.

Footprint Reachability Advertisement (FPR-Adv): MP-BGP Message used by a CDN in order to advertise or withdrawn Footprint reachability information.

Capability Advertisement (CAP-Adv): MP-BGP Message used by a CDN in order to advertise or withdrawn capability information.

3. CDNI Mesh and MP-BGP

CDNI enables CDNs to communicate in order to deliver content in a collaborative mode.

CDNI Mesh will make use of Multiprotocol-BGP (MP-BGP [RFC4760]) for the exchange of footprint, reachability and capability information.

We define a new Sub Address Family (CDNI-SAFI, TBD) for Address Families (AF) 1 (IPv4) and 2 (IPv6) and a set of new Network Layer Reachability Information (NLRI) that will carry either FPE, FPR or CAP advertisements. The NLRI will have a NLRI type determining the content (i.e.: FPE, FPR or CAP) so to distinguish advertisements.

The advantage of using a separate address family is to isolate CDNI information from regular BGP-4 Internet information so to not compromise in any way the security and reliability of the current BGP information exchange used for IP network layer routing.

The advantage of having separate FPE, FPR and CAP information is that a CDN needs not originate (and update) FPE information each time there’s a change in the way a CDN is connected to other CDNs. E.g.: if the complete Footprint information was to be exchanged between CDNs, it would consist of a very large amount of prefixes advertised (and re-advertised) each time a CDNI interconnection changes in the

CDNI Mesh.

When an existing CDNI connection is removed or when a new connection between two CDNs is established, the only advertisements that need to be updated are the ones concerning the FPR.

While footprint information is expected to be relatively stable, the CDNI Mesh (i.e.: the connectivity between CDNs) and the reachability from CDN to footprints may be impacted by network events. Also, the connectivity between CDNs may be affected by the CDN selection policy which may be modified relatively frequently.

CDNI FPR advertisements allow the CDNI Mesh to scale by adapting easily to topology changes. In fact, just a few number of FPR advertisements are used by each CDN which makes the CDNI MP-BGP scheme very scalable.

4. CDNI Information

CDNI Information includes:

Footprint (FP).

Footprint Element (FPE).

Footprint Reachability (FPR).

CDN Capabilities (CAP).

4.1. Footprint

Footprint represents the exhaustive set of prefixes a CDN is willing to serve. The set of prefixes may belong to one or more Autonomous Systems or may include subset of prefixes of one or more Autonomous Systems. In other words, there is no strict relationship between an Autonomous System and the CDN Footprint.

Example: if CDN-A delivers content to ISP-A users, then CDN-A footprint consists of all prefixes owned and connected to ISP-A.

4.2. Footprint Elements (FPE)

Footprint Elements refers to an arbitrary set of prefixes part of a CDN Footprint.

FPEs can be inferred from the BGP-4 Internet database. It is assumed that the CDN will have a BGP-4 feed with Internet prefixes that are

necessary in order for the CDNI Mesh to operate and from which it will be able to derive FPEs.

The CDN will maintain a FPE Database that is isolated from the regular BGP Internet database. The CDNI FPE database uses the CDNI-SAFI (TBD) MP-BGP address family.

FPEs can be implicit (inferred from BGP) or explicit (advertised).

Example 1: if CDN-A is willing to serve Autonomous System A and Autonomous System B users, then two footprints elements will be inferred from the BGP database in all CDNs: FPE-ASA and FPE-ASB. Then, CDN-A will advertise its reachability to FPEs FPE-ASA and FPE-ASB. In this example FPEs are inferred by all CDNs by just looking at the BGP-4 Internet Database.

Example 2: if CDN-B is willing to serve Autonomous System A users and a subset of Autonomous System B users, then CDN-B Footprint consist of two FPEs: Autonomous System A and the set of prefixes representing the subset of Autonomous System B users. Autonomous System 'A' FPE is inferred from BGP-4 Database while the subset of Autonomous System B users is an FPE that will be explicitly advertised by CDN-B to the CDNI Mesh.

The CDNI FPE Database includes FPEs inferred from the BGP-4 (Internet) database as well as FPE Advertisements originated by the CDN and/or received from other CDNI Mesh members.

4.3. Footprint Reachability (FPR)

Footprint Reachability refers to the way a CDN can reach to one or more FPEs. FPR consists of advertisements originated by each CDN for each of its FPE and advertised to the CDNI Mesh (through MP-BGP Messages).

FPR Information is stored in the CDNI FPR Database which contains the information originated by the CDN and the information received from the other CDNs in the CDNI Mesh.

4.4. CDN Capability Information (CAP)

Each CDN advertises to the CDNI Mesh the set of its capabilities. The CDN originates a CDN Capabilities message (CAP) message containing the attributes describing such capabilities.

Each CDN MUST originate a Capabilities message and each CDN will store in the CDNI Capabilities Database the set of Capabilities messages received from other members of the CDNI Mesh.

5. CDNI Functional Components

This section describes the CDNI functional components (databases, messages, information sets) stored and exchanged between CDNs and related to CDN Footprints and Capabilities. This document proposes the use of Multiprotocol-BGP for exchanging the CDNI information. The details of MP-BGP encodings are described in Section 7.

5.1. CDNI Databases

This section describes the set of databases defined for the propagation of Fooptprint and Capability information across the CDNI Mesh. The goal of this section is to explain the different information elements that are required in order to properly determine FPE, FPR and CAP information. This section is not to be intended as an implenentation description. An implementation may combine or merge all data into single or multiple databases.

CDNI Footprint Elements Database contains FPEs that are either derived from BGP-4 Internet table or received from other CDN through CDNI Footprint Elements Advertisements.

CDNI Footprint Reachability Database contains the advertisements originated by the CDN and received from other CDNs and that describe how each CDN can reach its Footprint Elements.

CDNI Capability Database contains the set of CDN capabilities advertised by each CDN.

5.1.1. CDNI Footprint Elements Database

FPE Database contains the FPEs that a CDN has inferred from the BGP-4 Internet database as well as FPEs that have been explicitly advertised by other CDNs.

FPE Database includes prefixes and their attributes as known in the BGP-4 Internet Database (e.g.: prefix, AS_PATH, MED, Communities, etc) in addition to CDNI specific attributes that are defined in following sections.

FPE Database also contains the FPEs that have been explicitly advertised by remote CDNs.

It is expected that for most of CDNs, the majority of the FPE information is inferred from BGP-4 Internet Database so to reduce significantly the advertisement of FPEs.

5.1.2. CDNI Footprint Reachability Database

FPR Database contains the set of FPEs each CDN claimed reachability to. This includes:

The set of FPEs this CDN can reach.

The set of FPEs other CDNs have advertised their reachability to.

The FPR Database is populated by the FPR Advertisement messages originated and sent by each CDN participating into the CDNI Mesh.

5.1.3. CDNI Capability Database

Capability (CAP) Database contains the set of capabilities advertised by each CDN.

The CAP Database is populated by the CAP Advertisement messages originated and sent by each CDN participating into the CDNI Mesh.

5.2. CDNI MP-BGP Messages

We define three new MP-BGP messages for the advertisement of CDNI Address Familiy NLRIs:

Footprint Element Advertisement (FPE Advertisement) describing an FPEs and its attributes.

Footprint Reachability Advertisement (FPR Advertisement) describing how a CDN can reach one or more FPEs.

Capability Advertisement (CAP Advertisement) describing CDN capabilities.

The propagation of CDNI Messages throughout the CDNI Mesh is done according to standard BGP rules and the inter-CDN connectivity will be reflected in the BGP attributes, e.g.: AS_PATH will describe the different CDNs the advertisement traversed during its propagation in the CDNI Mesh thus describing the inter-CDN connectivity.

5.2.1. CDNI MP-BGP Footprint Element Advertisement

FPE Advertisement is originated by a CDN willing to group prefixes it can reach under a unique identifier (identifying the FPE). The message is originated and sent out by the CDN and propagated throughout the CDNI Mesh according to MP-BGP propagation rules.

The FPE Advertisement includes CDNI NLRI, representing the FPE

address space, and attributes among which the identifier of the FPE.

5.2.2. CDNI MP-BGP Footprint Reachability Advertisement

In the CDNI Mesh, each CDN MUST advertise the different Footprint Elements it reaches. The CDN maintains a FPE Reachability Database with entries describing its connectivity to Footprints Elements.

The propagation of CDN Connectivity advertisements throughout the CDNI Mesh is done according to standard BGP rules and the inter-CDN connectivity will be reflected in the BGP attributes (e.g.: AS_PATH will describe the different CDNs the advertisement traversed during its propagation in the CDNI Mesh thus describing the inter-CDN connectivity).

5.2.3. CDNI Capability Advertisement

Each CDN MUST advertise information describing its capabilities. This is done in the CAP dvertisement message.

5.3. CDNI Information Elements (Attributes)

This section describes the different information elements contained in the CDNI Databases.

5.3.1. CDN Identifier

CDN_Identifier uniquely identifies the CDN within the CDNI Mesh. CDN Identifier is exchanged inside the CDNI (MP-BGP) messages between CDNs.

5.3.2. Footprint Element Identifier

The role of the Footprint Element Identifier is to group all prefixes part of the same FPE under a unique identifier. This allows a CDN to claim reachability to the FPE by just specifying the FPE Identifier rather than each individual prefix of the footprint.

When inferring the CDNI footprint information from the regular BGP-4 Internet database, the FPE Identifier is derived from the first ASN in the AS_PATH of the prefix, i.e.: the AS of origin of the prefix.

FPE Identifiers are present in both FPE Advertisement and FPR Advertisement. In the FPE Advertisement it is used to arbitrarily group prefixes under the same Identifier while in the FPR Advertisement, the FPE Identifier is used in order to describe which FPEs a CDN can reach.

5.3.3. Reachable Footprint Element

Reachable FPE contains the FPE Identifier of each FPE the CDN claim reachability to. Multiple instances of Reachable FPE are allowed in the same FPR Advertisement message. FPE Identifier is assigned to the FPE by the CDN advertising the FPE (in the FPE Advertisement Message). FPE Identifier contains an AS number when the FPE has been inferred from the BGP database.

5.3.4. Origin_AS_PATH

This attribute contains the prefix AS_PATH value that is present in the BGP-4 Internet Database of the CDN originating the FPR advertisement.

Origin_AS_PATH aims to describe how the CDN can reach a given FPE and is present only in FPR advertisements.

5.3.5. CDN Capabilities

Capabilities are expressed in a set of attributes the CDN inserts in the Capability Advertisement message.

5.4. CDNI MP-BGP NLRIs

We define three types of CDNI NLRI: FPE-NLRI, FPR-NLRI and CAP-NLRI

When referred to FPE, the NLRI contains either an IPv4 or IPv6 prefix.

In some cases, a given FPE prefix is reachable by multiple CDNs. MP-BGP, according to BGP selection rule, will allow to select only one FPE and therefore information about FPE being multi-reachability through different CDNs may be lost. In order to prevent this, the CDNI NLRI contains a CDNI FPE Distinguisher so to make two identical prefixes different from a BGP selection perspective. The behavior is similar to what defined in [RFC4364].

FPR NLRI consists of one or more FPE-Identifiers representing the FPEs the CDN can reach.

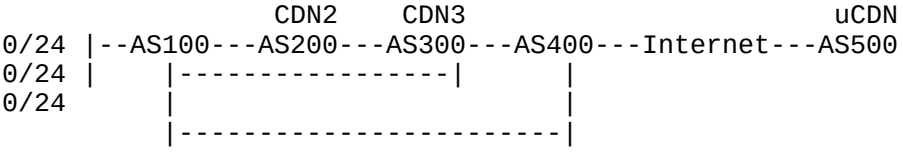
CAP-NLRI contains a IPv4 or IPv6 prefix (called the CAP-Prefix) beloning to the CDN address space originating the CAP Advertisement Message. Capabilities will be avertised using attributes to the CAP-prefix.

6. CDNI Example

This section illustrates an example of CDNI Mesh with the set of databases and messages that are exchanged between CDNs.

6.1. Topology

The following topology is used:



CDN2, CDN3 and uCDN participate into the CDNI Mesh.

CDNI Mesh includes following MP-BGP Sessions: CDN2-CDN3 and CDN3-uCDN. Note that there’s no MP-BGP session between CDN2 and uCDN.

AS100 and AS400 have no CDN.

AS100 advertises all prefixes to AS300 and AS400.

AS100 advertises only prefix 1.1.1.0/24 and 3.3.3.0/24 to AS200.

6.2. CDN2 Footprint Element Advertisements

CDN2 originates following Footprint Elements Advertisements:

```
        Prefix CDN2:1.1.1.0/24
           AS_PATH: {CDN2}
           Footprint-Element Identifier: FP-CDN2

        Prefix CDN2:3.3.3.0/24
           AS_PATH: {CDN2}
           Footprint-Element Identifier: FP-CDN2
```

where:

Prefix CDN2:1.1.1.0/24 and Prefix CDN2:3.3.3.0/24 are the two prefixes advertised with the CDNI FPE Distinguisher corresponding to CDN2. The CDNI FPE Distinguisher is required so to distinguish between possible identical prefixes advertised by other CDNs.

AS_PATH contains the set of CDNs the update has traversed. It is mainly used for loop prevention in BGP in the CDNI Mesh.

Note: the AS numbers in the AS_PATH represent the CDNs, not the BGP autonomous systems.

Footprint-Element Identifier contains the identifier of the Footprint-Element and it is used by CDN2 when originating the Footprint Reachability Advertisement.

Note that uCDN and CDN3 do NOT originate any Footprint Element Message.

6.3. Footprint Element Database in each CDN

Footprint Elements Database in constructed by inferring all info from BGP database plus the Footprint Elements Advertisements received from within the CDNI Mesh.

In CDN2 the Footprint Elements Database is as follows:

```
       1.1.1.0/24; AS_PATH: {100}         ===\
       2.2.2.0/24; AS_PATH: {300, 100}    ====> inferred from
       3.3.3.0/24; AS_PATH: {100}         ===/   BGP table

        CDN2:1.1.1.0/24
          AS_PATH: {}
          Footprint-Element Identifier: FP-CDN2

        CDN2:3.3.3.0/24
          AS_PATH: {}
          Footprint-Element Identifier: FP-CDN2
```

In CDN3 the Footprint Elements Database is as follows:

```
1.1.1.0/24; AS_PATH: {100}         ===\
2.2.2.0/24; AS_PATH: {100}         ====> inferred from
3.3.3.0/24; AS_PATH: {100}         ===/    BGP table

CDN2:1.1.1.0/24
  AS_PATH: {CDN2}
  Footprint-Elements Identifier: FP-CDN2

CDN2:3.3.3.0/24
  AS_PATH: {CDN2}
  Footprint-Element Identifier: FP-CDN2
```

In uCDN the Footprint Elements Database is as follows:

```
1.1.1.0/24; AS_PATH: {<internet>, 400, 100} ===\  inferred
2.2.2.0/24; AS_PATH: {<internet>, 400, 100} ====>   from
3.3.3.0/24; AS_PATH: {<internet>, 400, 100} ===/  BGP table

CDN2:1.1.1.0/24
  AS_PATH: {CDN3, CDN2}
  Footprint-Element Identifier: FP-CDN2

CDN2:3.3.3.0/24
  AS_PATH: {CDN3, CDN2}
  Footprint-Element Identifier: FP-CDN2
```

The Footprint-Elements Database contains prefixes inferred from the BGP database and the prefixes received through Footprint-Element Advertisements.

Footprint-Element prefixes derived from BGP database do not require a CDNI FPE Distinguisher.The way an implementation stores inferred FPE information (with or without CDNI FPE Distinguisher is out of the scope of this document.

The prefixes received by FPE Advertisements need a CDNI FPE Distinguisher because multiple CDNs may advertise the same set of prefixes and we need to preserve all from selection rules.

6.4. Footprint Reachability Advertisements

Each CDN will advertise its reachability to Footprint Elements:

CDN2 Footprint Reachability Advertisement:

```
        CDN-Identifier: CDN2
        AS_PATH: <will contain the AS_PATH representing the path
                  taken by Footprint-Reachability advertisement in
                  the CDNI Mesh>

        Reachable Footprint-Element: AS200
        Origin_AS_PATH: {}

        Reachable Footprint-Element: FP-CDN2
        Origin_AS_PATH: {100}

        Where:

        Origin_AS_PATH describes the BGP AS_PATHs as known in CDN2.
```

CDN2 advertises reachability to two sets of prefixes: one consists of all prefixes in AS200 and another set consists of prefixes he received from AS100 (and that he previously sent with Footprint-Element Advertisements).

```
CDN3 Footprint Reachability Advertisement:
       CDN-Identifier: CDN3
       AS_PATH: <will contain the AS_PATH representing the path
                 taken by Footprint-Reachability advertisement in
                 the CDNI Mesh>

       Reachable Footprint-Element: AS300
       Origin_AS_PATH: {}

       Reachable Footprint-Element: AS100
       Origin_AS_PATH: {200, 100}

       Where:

       Origin_AS_PATH describes the BGP AS_PATHs as known in CDN3.
```

CDN3 advertises reachability to two sets of prefixes: AS100 and AS300.

Each CDN advertises:

- CDNI Identifier.
- Set of reachable Footprint-Elements (i.e.: a set of FPE Identifiers) with their corresponding Origin_AS_PATH.
- The AS_PATH representing the BGP connectivity within the CDNI Mesh.

6.5. Capability Advertisements

Each CDN originates a CAP Advertisement messages carrying information about its capabilities. Each CDN will advertise:

A prefix owned by the CDN (i.e.: part of the CDN address space).

Its CDN Identifier.

A set of attributes describing the CDN capabilities (encoded as BGP Extended Community Attributes).

```
CDN2 Capability Advertisement:
       CDN-Identifier: CDN2
       Communities (std/ext): representing CDN2 capabilities
       AS_PATH: <will contain the AS_PATH representing the path
                 taken by Footprint-Reachability advertisement in
                 the CDNI Mesh>
       NLRI: CDN2-Prefix

CDN3 Capability Advertisement:
       CDN-Identifier: CDN3
       Communities (std/ext): representing CDN capabilities
       AS_PATH: <will contain the AS_PATH representing the path
                 taken by Footprint-Reachability advertisement in
                 the CDNI Mesh>
       NLRI: CDN3-Prefix
```

6.6. FPE, FPR and CAP Databases in uCDN

uCDN has the two following databases:

```
   Footprint-Elements Database:
       1.1.1.0/24
           AS_PATH: {<internet>, 400, 100} ==\
       2.2.2.0/24                          ===\
           AS_PATH: {<internet>, 400, 100} ====> inferred from
       3.3.3.0/24                          ===/    BGP table
           AS_PATH: {<internet>, 400, 100} ==/

       CDN2:1.1.1.0/24
           AS_PATH: {CDN3, CDN2}
           Footprint-Element Identifier: FP-CDN2
       CDN2:3.3.3.0/24
           AS_PATH: {CDN3, CDN2}
           Footprint-Element Identifier: FP-CDN2
```

```
      Footprint-Reachability Database:
          CDN-Identifier: CDN2
              AS-PATH: {300, 200}

              Reachable Footprint-Element: AS200
              Origin_AS_PATH: {}

              Reachable Footprint-Element: FP-CDN2
              Origin_AS_PATH: {100}

          CDN-Identifier: CDN3
              AS-PATH: {300}

              Reachable Footprint-Element: AS300
              Origin_AS_PATH: {}

              Reachable Footprint-Element: AS100
              Origin_AS_PATH: {200, 100}

      Capability Database:
   CDN-Identifier: CDN2
          Communities (std/ext): representing CDN2 capabilities
          AS_PATH: <will contain the AS_PATH representing the path
                    taken by Footprint-Reachability advertisement in
                    the CDNI Mesh>
          NLRI: CDN2-Prefix

   CDN-Identifier: CDN3
          Communities (std/ext): representing CDN capabilities
          AS_PATH: <will contain the AS_PATH representing the path
                    taken by Footprint-Reachability advertisement in
                    the CDNI Mesh>
          NLRI: CDN3-Prefix
```

The Footprint-Reachability and Capabilities databases also contains the AS_PATH representing the MP-BGP connectivity in the CDNI Mesh and the path taken by Footprint-Reachability Advertisements. It is anyway mandatory in BGP for message loop prevention.

6.7. Request Routing for user 2.2.2.2

The following workflow is used for a request coming from user 2.2.2.2:

User with address 2.2.2.2 sends a content request to uCDN.

uCDN does a lookup in Footprint-Elements Database for address 2.2.2.2 and finds following entry:

```
         2.2.2.0/24
              AS_PATH: {<internet>, 400, 100}
```

uCDN does a lookup in Footprint-Reachability Database and look for a CDN claiming connectivity to AS100. It finds following entry:

```
    CDN-Identifier: CDN3
        AS-PATH: {300}

        Reachable Footprint-Element: AS300
        Origin_AS_PATH: {}

        Reachable Footprint-Element: AS100
        Origin_AS_PATH: {200, 100}
```

uCDN can select CDN3 as downstream CDN.

6.8. Request Routing for user 1.1.1.1

The following workflow is used for a request coming from user 1.1.1.1:

User with address 1.1.1.1 sends a content request to uCDN.

uCDN does a lookup in Footprint-Elements Database for address 1.1.1.1 and finds following entries:

```
    1.1.1.0/24
        AS_PATH: {<internet>, 400, 100}
```

and

```
    CDN2:1.1.1.0/24
        AS_PATH: {CDN3, CDN2}
        Footprint-Element Identifier: FP-CDN2
```

uCDN prefers the entry originated by a Footprint-Element Advertisement and looks into the Footprint-Reachability Database for a CDN claiming connectivity to Footprint-Element Identifier: FP-CDN2. It finds following entries:

```
    CDN-Identifier: CDN2
        AS-PATH: {300, 200}

        Reachable Footprint-Element: AS200
        Origin_AS_PATH: {}

        Reachable Footprint-Element: FP-CDN2
        Origin_AS_PATH: {100}
```

uCDN can select CDN2 as downstream CDN

7. CDNI MP-BGP Encodings

This section describes the CDNI MP-BGP messages with their attributes and NLRI encodings details.

7.1. CDNI Attributes

CDNI uses standard BGP attributes in both FPE and FPR advertisements in addition to the following newly defined attributes:

FPE_Identifier is present in FPE advertisements (when identifying an FPE as a set of prefixes) and FPR advertisements (when identifying a reachable FPE).

CDN_Identifier is present in both FPE and FPR advertisements).

Origin_AS_PATH is present in FPR advertisements.

CDN Capabilities is present in CAP advertisements.

7.1.1. CDNI Identifier

CDN_Identifier is a mandatory transitive attribute consisting of an extended community (type TBD) uniquely identifying the CDN within the CDNI Mesh. CDNI_Identifier must be unique within the CDNI Mesh. The recommended method is to use (as for the FPE_Identifier) the AS or IP Address format as described in[RFC4360].

7.1.2. FPE Identifier

FPE Identifier is a mandatory and transitive attribute consisting of an extended community (type TBD) containing a Footprint Element identifier. The FPE_Identifier uniquely identifies the FPE within the CDNI Mesh. The format of the FPE_Identifier encodings should ensure uniqueness of values. The recommended method for encoding FPE_Identifier is to use the format as defined in RFC4360 and use either the AS or an IP address owned by the CDN. This method, and the extended community format, allows a CDN to create multiple unique FPE_Identifiers.

7.1.3. Origin_AS_PATH

Origin_AS_PATH is an optional transitive attribute having the same format as AS_PATH attribute and containing the AS_PATH value of the Footprint Element known in the CDN originating the FPR advertisement.

Origin_AS_PATH must be left unchanged and has the same format as AS_PATH BGP attribute as defined in [RFC4271].

7.1.4. Capabilities

Capabilities is a mandatory and transitive attribute consisting of an extended community (type TBD) containing a value representing a given capability (TBD). More than one Capability attribute is allowed so to allow a CDN to advertise multiple capabilities in the same message. The encodings of Extended Communities is defined in: [RFC4360].

7.2. CDNI Messages

Three messages are used: FPE Advertisement, FPR Advertisement and CAP Advertisement.

7.2.1. FPE Advertisement

Footprint Element Advertisement message is used by a CDN in order to advertise a Footprint Element that can't be inferred from the BGP-4 Internet Database and that needs explicit advertisement.

The FPE Advertisement consists of one or more prefixes that a CDN reaches. The FPE Advertisement scope is not to describe how the CDN can reach the FPE but rather what are the prefixes forming the FPE. The FPE Advertisement message should contain following information:

CDN_Identifier: uniquely identifies the CDN originating the message. It is mandatory and transitive attribute. CDN_Identifier format is described in Section 7.1.

Footprint-Element Identifier: a number uniquely identifying the FPE as defined in Section 7.1. FPE-Identifier is encoded as an extended community. It is mandatory and transitive.

Standard BGP attributes such as AS_PATH, Communities, Local_Preference, MED, etc. are also used in the FPE Advertisement Messages.

NLRI: the set of prefixes (address and mask) of the FPE.

A CDNI Distinguisher is prependd to each prefix of the FPE. The CDNI Distingsuiher allows multiple CDNs to advertise the same prefix in their FPE Advertisement. Each CDN must use a unique number. The recommended method for assigning RDs is to use the CDN AS number or an IP address owned by the CDN.

7.2.2. FPR Advertisement

Footprint Reachability Advertisement message is used by a CDN willing to advertise the FPEs it can reach. The FPR Advertisement consists of a set of FPE-Identifiers and their attributes. Following information should be present in the FPR Advertisement message:

CDN_Identifier: uniquely identifies the CDN originating the message. It is mandatory and transitive attribute. CDN_Identifier format is described in Section 7.1.

Reachable Footprint-Element: one or more AS numbers and/or FPE_Identifier representing the reachable FPEs. Reachable_FPE is encoded as a set of extended communities. At least one must be present in the message and the attribute is transitive. FPEs that have been inferred from the BGP4 Internet database are identified through their AS number and FPEs that have been explicitly advertised are identified by their FPE_Identifier.

Standard BGP attributes such as AS_PATH, Communities, Local_Preference, MED, etc. are also used in the FPR Advertisement Messages.

Origin_AS_PATH. This attribute is related to an FPE and contains the value of the BGP AS_PATH of the FPE in the CDN originating the FPR message. When used, it implies all prefixes of the same FPE share the same BGP AS_PATH in the CDN originating the FPR message.

NLRI: Reachable Footprint-Element: one or more FPE Identifiers representing the reachable FPEs. Reachable FPE is encoded in the CDNI NLRI. At least one must be present in the message. FPEs that have been inferred from the BGP4 Internet database are identified through their AS number and FPEs that have been explicitly advertised are identified by their FPE Identifier.

7.2.3. CAP Advertisement

Capability Advertisement message is used by a CDN in order to advertise its capabilities. Capabilities are encoded as a set of BGP Extended Communities. The CAP Advertisement Message contains:

A CDN_Identifier: uniquely identifies the CDN originating the message. It is mandatory and transitive attribute. CDN_Identifier format is described in Section 7.1.

One or more Extended Communities attributes describing capabilities.

Standard BGP attributes such as AS_PATH, Communities, Local_Preference, MED, etc. are also used in the CAP Advertisement Messages for policy reasons.

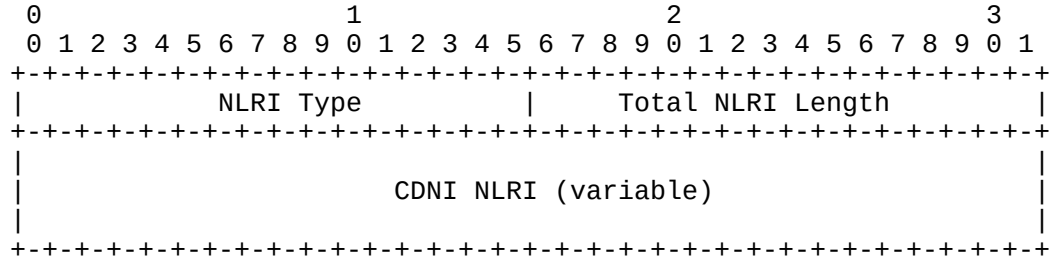
A CDN Prefix: a NLRI encoded prefix (IPv4 or IPv6) owned by the CDN (i.e.: a prefix that is part of the address space of the CDN).

7.3. CDNI NLRI

Multiprotocol-BGP defines two attributes, MP_REACH and MP_UNREACH, that are BGP's containers for carrying opaque information. Each CDNI NLRI describes either a FPE or a FPR.

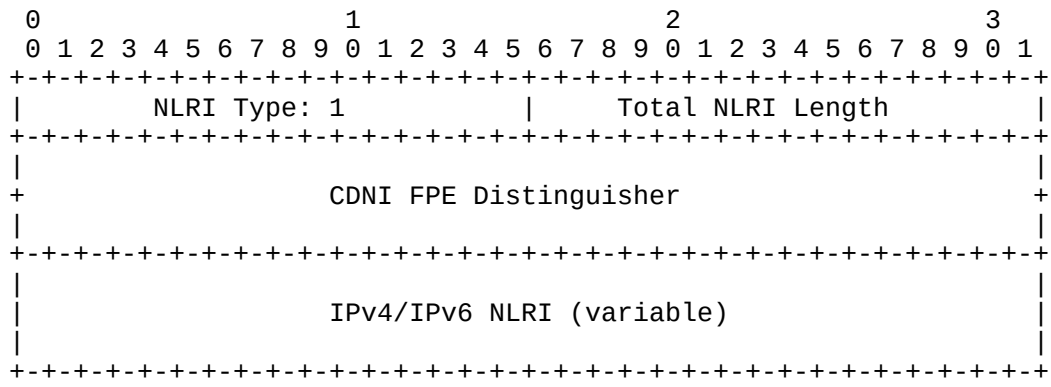
All FPE and FPR information SHALL be encoded using AFI 1 or 2 (ipv4 or ipv6) with a TBD (CDNI) SAFI.

In order for two BGP speakers to exchange CDNI NLRI, they MUST use BGP Capabilities Advertisement to ensure that they both are capable of properly processing CDNI NLRI. This is done as specified in [RFC4760], by using capability code 1 (multi-protocol BGP), with the AFI: 1 or 2, SAFI: TBD.



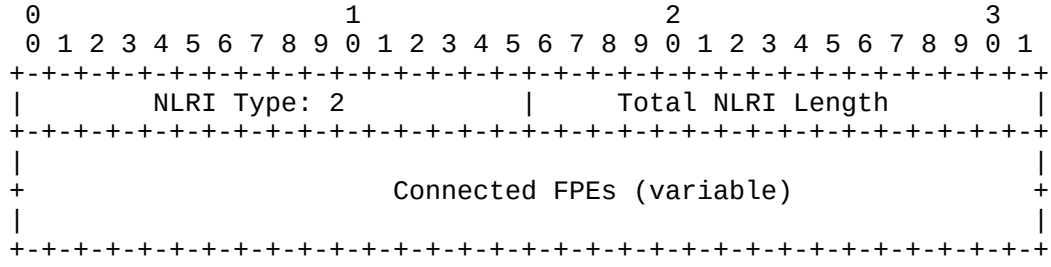
where:
NLRI Type: 1 (FPE)
2 (FPR)
3 (CAP)

7.3.1. CDNI-NLRI Type 1: FPE Advertisement



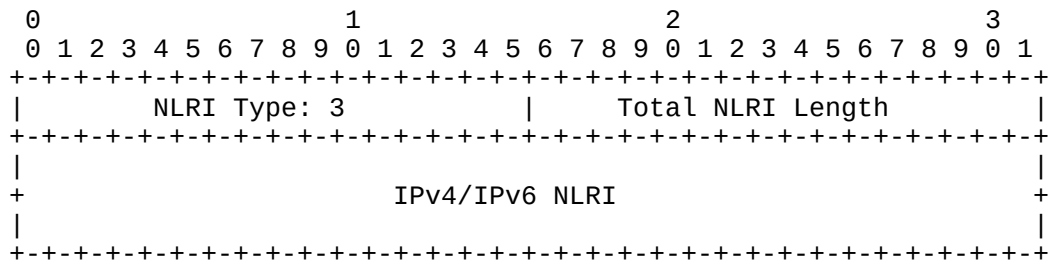
where:
NLRI Type: 1 specifies the FPE Advertisement
CDNI FPE Distinguisher: 8 octets
NLRI: IPv4 or IPv6 prefix

7.3.2. CDNI-NLRI Type 2: FPR Advertisement



where:
NLRI Type 2 specifies the FPR Advertisement
Connected FPEs: set of FPE Identifiers representing the FPEs the CDN can reach.

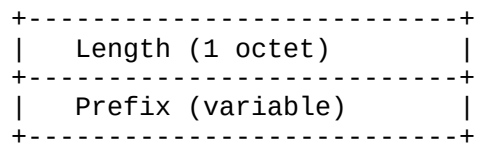
7.3.3. CDNI-NLRI Type 3: CAP Advertisement



where:
NLRI Type 3 specifies the CAP Advertisement
NLRI: IPv4 or IPv6 prefix

7.3.4. NLRI Encoding

The Network Layer Reachability Information contains one or more IPv4 or IPv6 prefixes, according to AFI value, and is encoded as one or more 2-tuples of the form <length, prefix>, whose fields are described below:



where:
Length: the Length field indicates the length in bits of the IP address prefix. A length of zero indicates a prefix that matches all IP addresses (with prefix, itself, of zero octets).

Prefix: the Prefix field contains an IP address prefix, followed by enough trailing bits to make the end of the field fall on an octet boundary. Note that the value of the trailing bits is irrelevant.

In the FPE Advertisement message, the NLRI contains the set of prefixes part of the FPE to be advertised.

In the CAP Advertisement message, the NLRI contains a prefix ownded by the CDN.

It has to be noted that a CDN may use multiple FPR messages for advertising reachability to multiple FPEs. In such cases multiple NLRI prefixes would be needed in order not to create collisions in the BGP selection process (that would select only one NLRI message among the ones having identical NLRI prefixes).

8. Compliance with CDNI Requirements

[I-D.ietf-cdni-requirements] outlines the requirements for the solution and interfaces to be specified by the CDNI working group. This section identifies the relevant requirements from that document and discusses compliance by the solution proposed in this document.

[Editor's Note: Text is to be added when requirements-03 is available. This needs to discuss the requirements labeled R27, R28, R29 and R30 as of requirements-02].

9. IANA Considerations

none.

10. Security Considerations

This draft does not affect the BGP security model.

11. Acknowledgements

The authors would like to thank Steven Luong, Manish Bhardwaj and Bruce Davie for their contributions.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, February 2006.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, January 2007.

12.2. Informative References

[I-D.bertrand-cdni-use-cases]
Bertrand, G., Stephan, E., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-bertrand-cdni-use-cases-02 (work in progress), July 2011.

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-01 (work in progress), October 2011.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-03 (work in progress), June 2012.

[I-D.jenkins-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-jenkins-cdni-problem-statement-02 (work in progress), March 2011.

Authors' Addresses

Stefano Previdi (editor)
Cisco Systems, Inc.
Via Del Serafico 200
Rome 00191
IT

Email: sprevidi@cisco.com

Francois Le Faucheur
Cisco Systems, Inc.
Greenside, 400 Avenue de Roumanille
Sophia Antipolis 06410
FR

Email: flefauch@cisco.com

Jan Medved
Cisco Systems, Inc.
3700 Cisco Way
SAN JOSE, CA 95134
US

Email: jmedved@cisco.com

Allan Guillou
SFR
40-42 Quai du Point du Jour
Boulogne-Billancourt 92659
FR

Email: allan.guillou@sfr.com

Content Delivery Networks Interconnection
Internet-Draft
Intended status: Informational
Expires: April 21, 2012

J. Seedorf
NEC

October 19, 2011

ALTO for CDNi Request Routing
draft-seedorf-alto-for-cdni-00

Abstract

Network Service Providers (NSPs) are currently considering to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. The necessary interfaces for inter-connecting CDNs are currently being defined in the Content Delivery Networks Interconnection (CDNi) WG. This document focusses on the Request Routing Interface of CDNi, and more specifically on how the solutions currently being defined in the Application Layer Traffic Optimization (ALTO) WG can improve CDNi request routing. The overall intention behind this document is to foster discussions (in the CDNi as well as in the ALTO WG) regarding a) if, b) how, and c) under what conditions ALTO can be useful to optimize CDNi request routing. As basis for this discussion, this document provides concrete examples of how ALTO can be integrated within CDNi request routing. The examples in this document are based on the use cases and examples currently being discussed in the CDNi WG.

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 April 21, 2012.

Copyright Notice

Table of Contents

1. Introduction

Many Network Service Providers (NSPs) are currently considering or have already started to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. Content Delivery Networks Interconnection (CDNi) has the goal of standardizing protocols to enable such interconnection of CDNs [refs.cdniproblemstatement].

The CDNi problem statement envisions four interfaces to be standardized within the IETF for CDN interconnection [refs.cdniproblemstatement]:

- CDNI Request Routing Interface
- CDNI Metadata Interface
- CDNI Logging Interface
- CDNI Control Interface

This document focusses solely on the CDNI Request Routing Interface. In particular, this document shows concrete examples of how ALTO [RFC5693] can be integrated in CDNi request routing. The goal of this document is to show in what cases ALTO can benefit CDNi request routing, giving concrete examples and explaining how ALTO improves CDNi request routing in each of these examples. The examples used in this document are based on the use cases and request routing proposals currently being discussed in the CDNi WG [refs.cdniusecases] [refs.cdnistrawman] and in the ALTO WG [refs.altocdn]. The overall rationale of this document is to foster discussions (in the CDNi as well as in the ALTO WG) regarding a) if, b) how, and c) under what conditions ALTO can be useful to optimize CDNi request routing.

Throughout this document, we use the terminology for CDNi defined in [refs.cdniproblemstatement].

2. CDNi Request Routing

The main purpose of the CDNI Request Routing Interface is described in [refs.cdniproblemstatement] as follows: "The CDNI Request Routing interface enables a Request Routing function in an upstream CDN to query a Request Routing function in a downstream CDN to determine if the downstream CDN is able (and willing) to accept the delegated content request and to allow the downstream CDN to control what the upstream Request Routing function should return to the User Agent in the redirection message". On a high level, the scope of the CDNI Request Routing Interface therefore contains two main tasks:

- A) Determining if the downstream CDN is willing to accept a delegated content request
- B) Redirecting the content request coming from an upstream CDN to the proper entry point or entity in the downstream CDN

3. Using ALTO within CDNi Request Routing

Application Layer Traffic Optimization (ALTO) is an approach for guiding the resource provider selection process in distributed applications that can choose among several candidate resources providers to retrieve a given resource. By conveying network layer (topology) information, an ALTO server can provide important information to "guide" the resource provider selection process in distributed applications. Usually, it is assumed that an ALTO server conveys information these applications cannot measure themselves [RFC5693].

Originally, ALTO was motivated by the huge amount of cross-ISP traffic generated by P2P applications [RFC5693]. Recently, however, ALTO is also being considered for improving the request routing in CDNs [refs.altocdn]. In this context, it has also been proposed to use ALTO for selecting an entry-point in a downstream NSP's network (see section 3.4 "CDN delivering Over-The-Top of a NSP's network" in [refs.altocdn]). Also, the CDNi problem statement explicitly mentions ALTO as a candidate protocol for "algorithms for selection of CDN or Surrogate by Request-Routing systems" [refs.cdniproblemstatement]. Yet, there have not been concrete proposals so far on how to use ALTO in the context of CDN interconnection. This document tries to close this gap by giving some examples on how ALTO could be used within CDNi request routing.

As explicitly being out-of-scope for CDNi [refs.cdniproblemstatement], the examples used in this document assume that ingestion of content or acquiring content across CDNs is not part of request routing as considered within CDNi standardization work. The focus of using ALTO (as considered in this document) is hence on request routing only, assuming that the content (desired by the end user) is available in the downstream CDN (or can be aquired by the downstream CDN by some means).

3.1. ALTO to simplify DNS-based Request Routing Redirection

If CDNi request routing is based on DNS, ALTO can potentially help to avoid one or more DNS resolution steps. For instance, Figure 1 shows a modified version of the high-level message sequence chart from Figure 5 of [refs.cdniframework] (note that this figure is similar to the high-level message sequence chart shown in Figure 3 of [refs.cdnistrawman]). In the original figure (i.e. Figure 5 in [refs.cdniframework]), the DNS server hosted by the upstream CDN (assumed to be the authoritative DNS server for the requested content), returns a DNS CNAME and NS record which essentially directs the end user to the request router of the downstream CDN. However, this redirection involves another DNS resolution for the request

router of the downstream CDN to be performed by the end user.

In the example using ALTO provided in Figure 1 of this document, the downstream CDN provides the upstream CDN an ALTO network (and corresponding cost) map by means of the ALTO protocol [refs.altoprotocol] (0). This ALTO map provides sufficient information for the upstream CDN to directly return a suitable IP-address for the CDN entry point in the downstream CDN (2).

In principle, using ALTO this way the downstream CDN provider would provide the decision on which delivery node is best by means of an ALTO network map to the upstream CDN provider. This enables the upstream CDN provider to directly return a suitable delivery node in the downstream CDN to the end user as a response to the initial DNS request received by the upstream CDN provider.

An imlicit assumption for the example in Figure 1 to work is that the CDN entry point for the downstream CDN only depends on the location of the end user. Using the cost map, the upstream CDN can determine the "best" entry point in the downstream CDN. If the "best" entry point depends also on the target domain ("cdn.csp.com" in the example), it becomes more tricky to make such information available to the upstream CDN by means of ALTO network map and cost map. One possible way to still use ALTO in this case would be for the downstream CDN to provide a different cost map for each Content Service Provider (CSP).

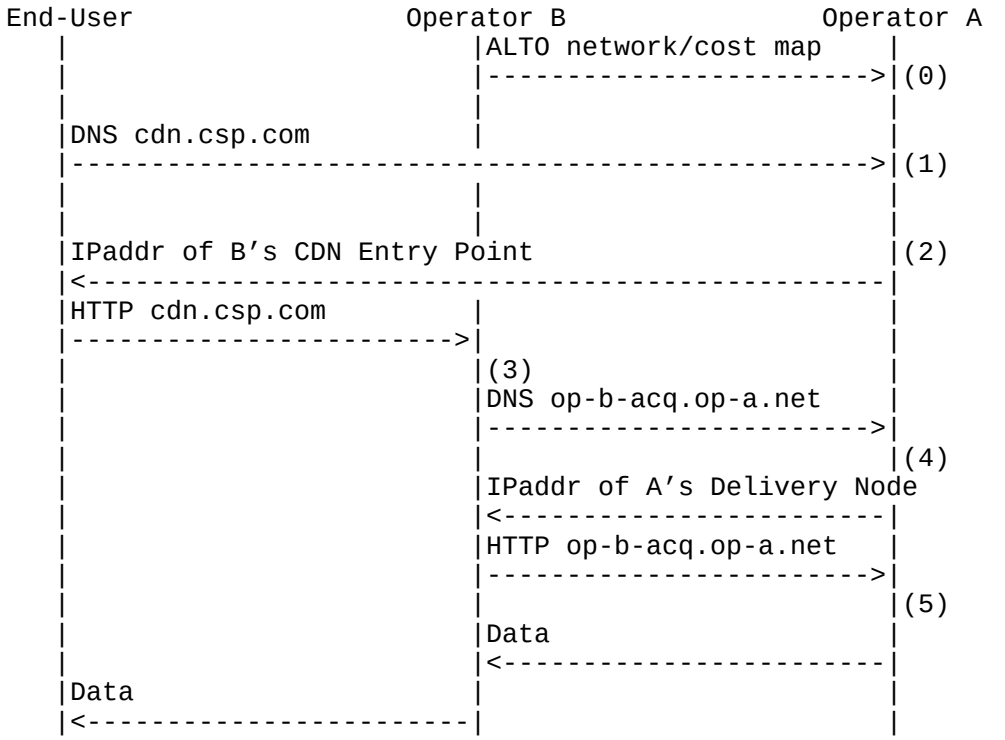


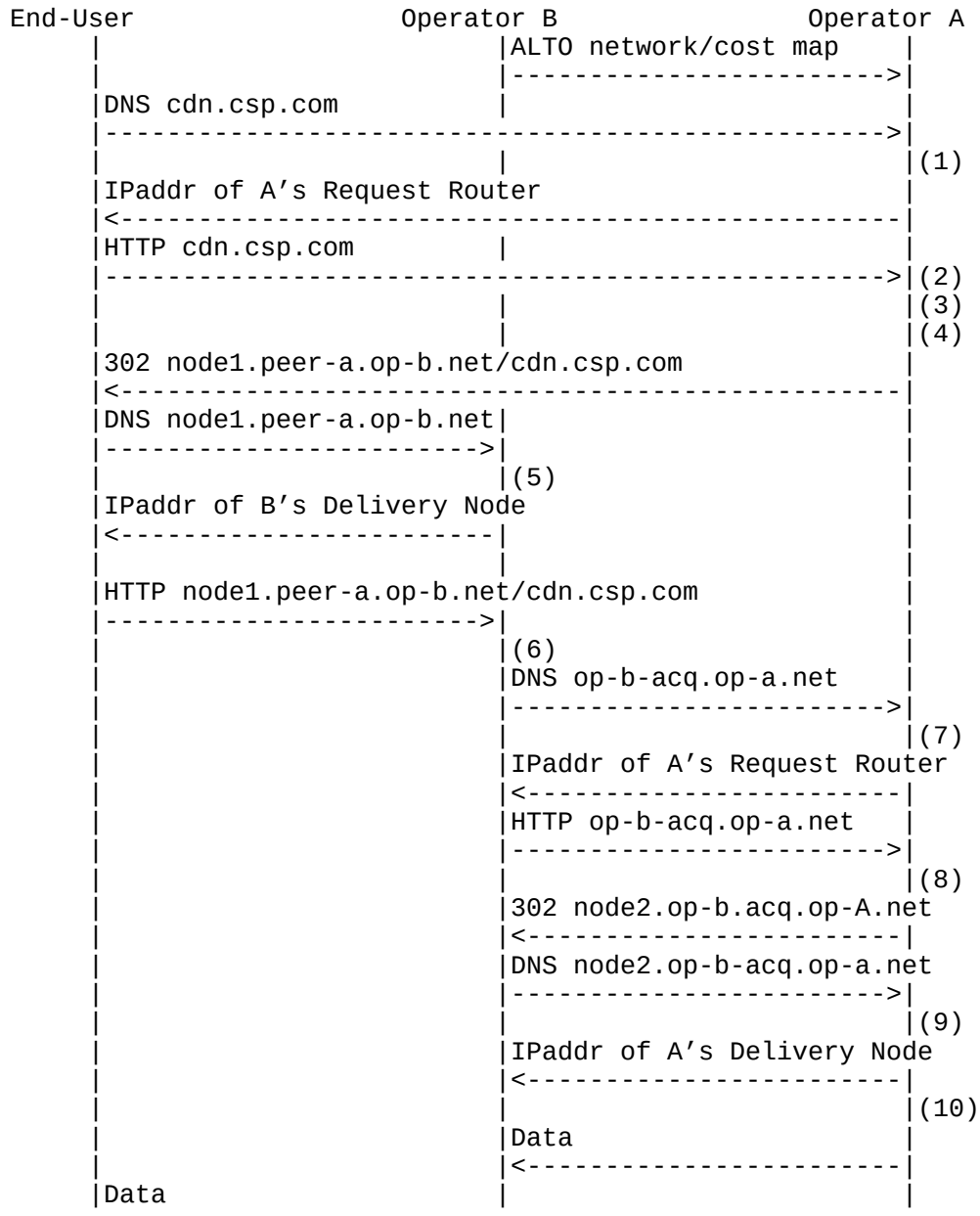
Figure 1 - ALTO within DNS-based redirection of request routing

3.2. ALTO to simplify http-Redirection for Request Routing

Similar to the example given in the previous subsection, ALTO can also help to reduce intermediate http redirection steps. Figure 2 shows a modified version of the the high-level message sequence chart from Figure 3 of [refs.cdniframework] (note that this figure is similar to the high-level message sequence chart shown in Figure 1 of [refs.cdnistrawman]). In this case, the ALTO maps provided in step (0) by the downstream CDN potentially enable the upstream CDN to directly return - as a response to an http request - the hostname of the suitable cache (delivery node) in the downstream CDN by means of 302-redirection (4). This use of ALTO would enable to avoid several http-302 redirections and DNS resolutions by the end user (compare with figure 3 of [refs.cdniproblemstatement], steps (2-4)).

Depending on how dynamically the actual "best" delivery node changes (from the downstream CDN's request routing perspective), it might only be meaningful to return to the end user the "best" entry point

or cluster within the downstream CDN. This would require an additional http-redirect by the downstream request router to the "best" actual cache. However, even in such a case ALTO can save one http-redirect and one DNS resolution at the end user, consequently speeding up the overall process of CDNi request routing.



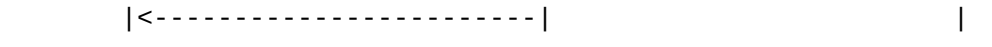


Figure 2 - ALTO within http-based redirection of request routing

3.3. ALTO to support Selection of Downstream CDN

ALTO could also help for the upstream CDN provider to select a proper downstream CDN provider for a given end user request. For instance, a network map provided by each of several candidate downstream CDNs could provide information to the upstream CDN provider regarding the geopgraphical coverage, the location of "surrogates", or similar. Future versions of this document will discuss this use case in more detail, and provide concrete examples how ALTO can be used for downstream CDN selection by the upstream CDN provider.

4. Security Considerations

Security Considerations will be discussed in a future version of this document.

5. Summary and Outlook

This document presented some examples on how ALTO can be used within CDNi Request Routing and argued why such use of ALTO is meaningful in certain cases. The intention of this document is to arouse discussions in the CDNi WG as well as the ALTO WG in order to find consensus on scenarios where ALTO is beneficial to CDNi request routing and to form agreement on how ALTO should be used in such cases within the CDNi request routing protocol. It is the intention to capture the outcome of such continuing discussions in future versions of this document.

6. Acknowledgements

Jan Seedorf is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.

7. Informative References

[refs.cdniusecases]
Bertrand, G., Stephan, E., Watson, G., Burbridge, T., and P. Eardley, "SPEERMINT Peering Architecture", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

[refs.cdniproblemstatement]
Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-00 (work in progress), September 2011.

[refs.altocdn]
Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and S. Previdi, "Use Cases for ALTO within CDNs", draft-jenkins-alto-cdn-use-cases-01 (work in progress), June 2011.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

[refs.cdnistrawman]
Peterson, L. and J. Hartman, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-peterson-cdni-strawman-01 (work in progress), May 2011.

[refs.cdniframework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[refs.altoprotocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-09 (work in progress), June 2011.

Author's Address

Jan Seedorf
NEC Laboratories Europe, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg 69115
Germany

Phone: +49 (0) 6221 4342 221
Email: jan.seedorf@neclab.eu
URI: http://www.neclab.eu

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 26, 2012

E. Stephan
G. Bertrand
F. Fieau
R. Pages
France Telecom Orange
October 24, 2011

metadata for CDNs Interconnection
draft-stephan-cdni-usecases-metadata-00

Abstract

There are use cases where the delivery of contents by a CDN on the behalf of another requires the exchange of extra information between these CDNs. This memo proposes a RESTful framework for the exchange of content distribution metadata between an upstream and a downstream CDN. Then it applies this framework to the use cases selected by te CDNi WG [I-D.ietf-cdni-use-cases] to identify relevant operations, objects and information elements of the CDNi metadata interface and to verify that the RESTful approach suits for these operations.

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 April 26, 2012.

Copyright Notice

Table of Contents

1. Introduction

There are use cases where the delivery of contents by a CDN on the behalf of another requires the exchange of extra information between these CDNs. This memo proposes a RESTful framework for the exchange of content distribution metadata between an upstream and a downstream CDN. Then it applies this framework to the use cases selected by te CDNi WG [I-D.ietf-cdni-use-cases] to identify relevant operations, objects and information elements of the CDNi metadata interface and to verify that the RESTful approach suits for these operations.

This document does not study all the metadata to be exchanged on the CDNi interface. It focuses on the specification of the exchange of content distribution metadata between an upstream CDN and a downstream CDN when the upstream CDN decides to distribute contents through this downstream CDN.

This draft compliments the works done by Ben in [I-D.jenkins-cdni-metadata] and Kevin in [I-D.ma-cdni-metadata].

1.1. Terminology

The document reuses terminology defined by CDNi WG documents [I-D.ietf-cdni-problem-statement] and [I-D.ietf-cdni-requirements].

The call flows of this document uses the following abbreviations for the interfaces (Control, Metadata, Request Routing, Logging) of the CDNi solution:

- Metadata interface: Mi
- Request Routing interface: RRi
- Content Acquisition interface: CAi
- Control interface: CTi

Content distribution metadata:

Information communicated by the upstream CDN to the downstream CDN that must be enforced by the downstream CDN to distribute contents on the behalf of the upstream CDN. The document uses the abbreviations 'CDmD'.

Push or Pull of metadata:

The upstream CDN associates the delegation of the distribution of contents with metadata to provide the downstream CDN with the

information needed to distribute the content according to the rules determined by the upstream CDN.

- A metadata is pushed on the CDNi Mi when the operation is initiated by the upstream CDN.

- A metadata is pulled on the CDNi Mi when the operation is initiated by the downstream CDN.

Time-to-service:

The delay between the request of a service and the delivery of the service (e.g. the delay between the selection of a VoD by an eyeball and the display of the first picture of this VoD).

information element (IE):

Piece of information of the information model.

Objects:

IE which can be individually operated through the metadata interface: created, read, modified and deleted.

Main objects:

IE which are addressable and can be individually created or deleted.

Ancillary objects:

Addressable IE which can be downloaded or modified individually.

Scalar:

IE that can not be individually addressed.

Sparse and dense mode:

The intensity of usage of a CDN interconnection may differ dramatically:

- A CDN interconnection is said ’sparse’ when dCDN rarely distribute contents on the behalf of uCDN.

- A CDN interconnection is said ’dense’ when dCDN intensively distribute contents on the behalf of uCDN.

2. Content Distribution Metadata Framework

This section presents existing metadata related to the distribution of contents then it defines the framework for exchanging metadata on the CDNi Mi interface.

2.1. Introduction to Content Distribution Metadata

Metadata is defined literally as 'data about data'. Practically it represents structured data about resources that help operating these resources. They may be stored in a file which captures the characteristics of a resource or dynamically generated like the sitemap of a WEB site. Metadata are designed according to the operational requirements of a domain. MPEG21, MPEG 7, TVAnyTime, DublinCore and Atom are notorious examples for the content domain.

Following are metadata directly tied to the content distribution domain.

2.1.1. HTTP Adaptive Streaming

HTTP adaptive streaming services distribute content in segments. The segments provide boundaries for performing rate adaptation. The segments are described as URL in manifest files.

There are several HTTP adaptive streaming:

- HTTP Live Streaming (HLS) [I-D.pantos-http-live-streaming] uses a hierarchy of m3u8 playlists pointing to individual segment files or other m3u8 playlists;

- Smooth Streaming [IIS-Smooth-Streaming] uses a set of XML files to describe the media source;

- HTTP Dynamic Streaming [Flash-Media-Manifest] uses the flash media manifest file XML format;

- Dynamic Adaptive Streaming over HTTP [MPEG-DASH] [ETSI-3GP-DASH-R10] also supports an XML manifest format;

These manifest files contain metadata about the organization of the content being delivered. This may include the location of the content as well as the order in which it is likely to be retrieved.

CDNs uses the CDNi metadata interface to exchange information on HAS streams to prepare their delivery. This encompasses the exchange of information about the delivery and the update of the manifest file.

2.1.2. IPTV CoD content distribution metadata

IPTV services distributes the technical information of the distribution of a content in metadata. They provide the terminal with the technical information for accessing to the content.

[DVB-IP-TV] specification includes the specification of the metadata for scheduling and for the controlling the regionalization of the distribution of a content.

Part 'A' of the specification of TV-Anytime metadata [TVA-CD-mD] specifies XML objects describing the parameters of a content distribution. 'ProgramLocationType' describes the content and where it must be distributed. 'ScheduleEventType' describes when the content must be distributed. These metadata are used in OIPF specifications. Before receiving a content an Open IPTV Terminal (OITF) receives technical information from the OIPF IPTV platform. The information is sent at the format of TV-Anytime XSD.

[I-D.thompson-cdni-atis-scenarios] presents the CoD service specified by ATIS [ATIS-0800042] . The metadata (namely Media Resource Metadata) are specified in a XML schema.

2.1.3. SNIA CDMI

CDMI [SNIA-CDMI-1.0] defines the functional interface that applications will use to create, retrieve, update and delete data elements.

2.2. Framework for exchanging CDmD

This section presents the framework.

2.2.1. RESTful design

In a CDNi interconnection dCDN provides the content distribution service to uCDN. Contents and Content Distribution metadata are going from uCDN to dCDN.

The framework is based on a RESTful model:

- dCDN is the server and uCDN is the client;

- The default protocol of CDNi Metadata interface is HTTP [RFC2616] over a secure transport layer like TLS 1.1 [RFC4346].

- uCDN uses the HTTP requests PUT, POST, GET, HEAD and DELETE methods to manage all the live cycle of the metadata;

- CDNi metadata interface does not define new methods than those of HTTP;

- The CDmD uploaded by uCDN are not accessible to other CDNs which interconnect with dCDN too.

2.2.2. Push of CDmD

CDmD operations occurs at various states of the distribution of a content:

- CDNi interconnection bootstrapping;

- Set up of a the delivery of a content or a group of content;

- Modification of the CDmD of a content or of a group of content;

- Reception of a request routing from an eyeball (before, during or after its redirection by uCDN to dCDN);

- Purge of a content;

Section 3 of the draft checks that the push mode covers the situations presented in the use cases selected by the WG CDNi. It allows uCDN to synchronize the CDmD operations with the CSP queries.

Discussion:

Pull mode might be used too but it does not cover any situations and comes at the expense of an increase of the time-to-service (see [I-D.stiemerling-cdni-routing-cons]) and of the burstiness of the signalling on the CDNi interfaces. Furthemore when uCDN has to reflect content distribution demands of its CSP it does not provide uCDN with a simple mechanism to send CDmD to dCDN. Consequently the usage of pull mode requires some push mode too and increases the metadata interface complexity.

2.2.3. Datamodeling Language

Existing content distribution metadata are generally specified in XML schema. As they correspond to information describing one content sent to an eyeball they do not correspond exactly to the one that should be exchanged by 2 CDNs:

- Content acquisition between 2 CDNs differs from content delivery, especially direct delivery. A content may consists in a single file from the acquisition perspective (e.g. zipped) while being delivered to the end-user in small files, e.g. VOD HAS

example;

- Scalability: CD mD are send mostly individually to eyeball terminals. An efficient CDNi metadata interface requires the grouping of the exchange of metadata between the 2 CDNs;

The data modeling is based on XML schema. Several languages are canditate for exchanging information.

- XML: Intensively used; XML come along with W3C XML Schema for specifying interfaces and XPATH to transform and selectively extract data out of XML document ;

- IETF YANG/NETCONF, RFC 6020; human readable, XML and RELAX HG interoperability;

- OASIS Relax Ng;

- JSON: Intensively used; human readable; Nothing equivalent to XSD schema and XPATH;

2.2.4. On-the-fly vs Batch CDmD Exchange

CDmD exchange (same applies for Content acquisition and purge) is said to be Batch when the preparation or the endding of the distribution of a content over the CDN interconnection is not timely correlated with the delivery of the content to the eyeball.

CDmD exchange is said to be on-the-fly when it is triggered by the arrival of a request routing query from an eyeball on the RRi of uCDN or dCDN.

Information elements:

On-the-fly actions are very demanding in terms of processing and signaling. A CDN may not support them or all of them. the capability object reflects the capability of a given CDN to support them:

- CDmD Exchange mode: the mode supported, either batch or on-the-fly;

- Content acquisition mode: the mode supported, either batch or on-the-fly;

- CDmD deletion mode: the mode supported, either batch or on-the-fly;

-CDmD purge mode: the mode supported, either batch or on-the-fly;

Discussion:

The list of flags above is not be exaustive. Overspecifying the capabilities (i.e. splitting the description of one capability in 2 flags) will not arm the CDNi performance. As an example CDmD Exchange mode can be split in 'CDmD' reception mode' (able to receive on the fly or not) and 'CDmD sending mode' (may send on the fly or not).

uCDN must exchange similar information to the dCDN must inform of its capabilities to.

2.2.5. Objects Extension

CDNs interoperate to distribute content services that may require specific metadata. Consequently each main object is extensible and includes per

design a field 'extension' for this purpose. GPX XML schema uses this approach (See Annex A). The extension field can be used to carry opaque information as requested by [I-D.ma-cdni-metadata].

Discussion

In GPX schema each field is extensible. Will all the objects of the CDNi Mi datamodel be extensible?

2.2.6. Operations

This section presents Mi operations triggered by uCDN (with effect in the dCDN).

2.2.6.1. Creation of Object

uCDN uses the HTTP method PUT to create metadata in main objects in dCDN.

on success dCDN returns 201 'Created'.

When dCDN detects a format error it returns an HTTP error 400 'Bad Request'

When dCDN does not support the metadata object received it returns the HTTP error 403 'Forbidden'.

When dCDN received a PUT method for the creation of a object that is

not supported it shall return the method 405 'Method Not Allowed'.

When dCDN receives an object that uCDN is not authorize to create it returns the HTTP error 401 'Unauthorized'.

2.2.6.2. Update of an object

uCDN uses the method POST to update an addressable object in dCDN.

on success dCDN returns 200 'OK'.

[edit] add error cases

2.2.6.3. Consultation of an Object

uCDN requests the value of an object using a HTTP GET method.

When the object exists and is addressable dCDN returns the value of object.

when the object does not exist uCDN returns 410 'Gone'

2.2.6.4. Deletion of an Object

uCDN uses the HTTP method DELETE to remove the CD metadata from the dCDN. dCDN removes the object and returns HTTP '201' OK.

The deletion of CDmD metadata is performed by content or by set of contents. As an example, individual deletion happens when a CoD content is removed from a catalogue, and a set of deletion happens when a CoD is stored in fragments corresponding to chapters.

Check of deletion:

uCDN checks the deletion with a HTTP GET asking for the object that have be deleted. dCDN should return 410 'Gone'.

Discussion:

The purge of the content is not in the scope of the MI interface. Nevertheless the deletetion of the CDmD of a content is coupled with the purge of the content.

2.2.7. Main Objects

The metadata interface is designed to operate main objects. It provides operations to fully (create, delete..) manage main objects, to modify objects but it does not support atomic management of scalar

parameters. As an example the WG may decide that the 'start of delivery' of a content metadata can not be modified atomically, i.e. without modifying the content object. It is up to the CDNi WG to determine the level of each information elements. Nevertheless for the sake of interoperability the number of operations has to be limited:

The data model is made of objects connected by the operations needed by the interconnection. A main object is an item that can be created and deleted independently of the others. As an example, geographic areas are complex objects that are managed individually.

The study made in section 3 identifies the following object organisation. It does not specify a datamodel. It gathers the information elements identified in section 3 to ease the reading and the specification of the call flow and of the operations:

Main object 'content' :

- A group of pieces of content.
- Operations:
  - uCDN creates a content;
  - uCDN adds a piece of content in a content;
  - uCDN deletes a piece of content from a content;
  - uCDN updates the parameters
- IEs:
  - content activation;
  - content deactivation;
  - expiration times;
  - cache invalidation and removal intervals;
  - source URL
  - backup source URL;

- auto_purge_delay;

- minimal_storage_duration: duration of the storage of a content;

- immediat_acquisition: flag requesting that dCDN must start the acquisition triggered on reception of the content metadata;

- 'extension'

Main object 'region' :

- standard administrative names of geographic area like country, region,... like defined ISO 3166-2.

- Operations:

  - uCDN gets the name of regions predefined by dCDN.

  - uCDN creates a logical region made of a subset of names predefined by uCDN;

  - uCDN gets, modifies, deletes a logical region.

  - uCDN gets the regions where dCDN supports on net delivery;

  - uCDN gets the network prefixes of a region where dCDN supports on net delivery;

- IEs

  - Names of geographic areas like countries, part of country, district, city...

  - on_net flag: indicates the support or not of on-net delivery : 'full', 'partial' or 'none';

  - on_net_geoIP;

  - Capacity:peak (unit, value), sustainable: (unit, value); duration (start, end);

  - 'extension'

Object 'Geoloc':

- Detailed network information on a region. Lookup information resolving the localisation of a host based on its network address.

CDNs embed geoIP database and are usually able to resolve geoIP localisation without exchanging Geolocalisation information. Nevertheless several use cases require the exchange of the addresses of the eyeball networks that a CDN covers.

- Operations:

  uCDN gets the Geoloc of a region.

- IEs

  - IP prefixes

  -'extension'

Main object 'dCDNcaps'

- Operations:

  uCDN gets dCDN capablities.

- IEs

  - URL_format;

  - token_format

  - regions: details of dCDN footprints. A set of names of geographic area like country, region,..

  - capacity

  - interconnection capability,

    - CDmD_exchange_mode: 'batch', 'on-the-fly' or 'both';

    - CDmD_deletion_mode: 'batch', 'on-the-fly' or 'both';

    - CDmD purge mode: 'batch', 'on-the-fly' or 'both';

    - Content acquisition mode: 'batch', 'on-the-fly or 'both'';

    - 'extension'

-'extension'

Main object 'uCDNcaps'

- Operations:

  uCDN creates and sets an uCDNcaps object on dCDN Mi interface.

- IEs

  - on_dCDN_failure_FQDN: FQDN of the uCDN RRi which handles the redirection on failure of the redirection by dCDN.

  backupCDN: FQDN of a content acquisition backup;

  - bursty_ interconnection: a flag saying that the interconnection is very bursty;

  -'extension'

2.3. Bootstrapping of the CDmD interface

This section makes the asumption that uCDN and dCDN have an technical agreement which defines the usage of dCDN resources by uCDN and exchanged bootstrapping information like RRi and CAi server addressesof whole CDNi bootstrapping on the control interface or manually.

The initialization of the Metadata interface includes the steps below :

1. dCDN grants uCDN with the right to connect, upload and download CDmD on its metadata interface;

2. dCDN initiates its capabilities objects according to the technical agreement it has with uCDN;

3. uCDN connects to dCDN CDmD server;

4. uCDN gets dCDN capabilities;

5. uCDN setups high level CDmD in dCDN;

6. Setup of objects needed during the duration of the interconnection (e.g. a group of content that may be modified but unlikely deleted);

Discussion:

The boundary between the information needed by the control interface and the Mi interface is not clear. Some IEs may be needed on both (e.g. the RRi server of a region).

2.4. Comparison of CDNi CDmD and SNIA/CDMI

Follwing is a list of points above which are already specified by the the CDMI interface [SNIA-CDMI-1.0]:

- Both are client/server and RESTful;
- Both requires metadata extensibility (section 16.2);
- CDMi reuses HTTP 1.1 primitives and error codes to implement the operations;
- Secure Transport;
- both must balance between XML and JSON;
- Same operations:
  - discovery of capabilities;
  - creation
  - deletion;
  - read;

Most of the aspects of the CDNi Mi are included in CDMI. A deeper insight is needed to determine if CDNi Mi can be specified as a subset of CDMI where uCDN uses the Data Storage Interface and dCDN acts in the role of providing a Data Storage Interface.

3. Metadata exchanged for CDNi use cases

[I-D.ietf-cdni-use-cases] presents realistic situations between 2 CDNs where the downstream CDN ingests and delivers contents on the behalf of the upstream CDN. This section studies the exchange of metadata for these situations. Each subsection presents the exchange of content distribution metadata for one use case.

Only one example of call flow is shown per use case. DNS steps are not represented to simplify the call flows. They are discussed in

[I-D.peterson-cdni-strawman].

3.1. Example of Initialization of the CDmD exchange

This section makes the asumption that uCDN and dCDN have an technical agreement which defines the usage of dCDN resources by uCDN. It gathers CDmD exchanges used by several call flows.

The initialization of the Metadata interface is as follow :

1. dCDN grants uCDN with the right to connect, upload and download CDmD on its metadata interface;

2. dCDN initiates its capabilities objects according to the technical agreement it has with uCDN;

3. uCDN connects to dCDN CDmD server;

4. uCDN gets dCDN capabilities;

5. uCDN setups high level CDmD in dCDN;

6. optionally, uCDN setups objects it is likely to need during the duration of the interconnection (e.g. a group of content that may be modified but unlikely deleted);

Following is an example of call flow for the initialization of the metadata interface.

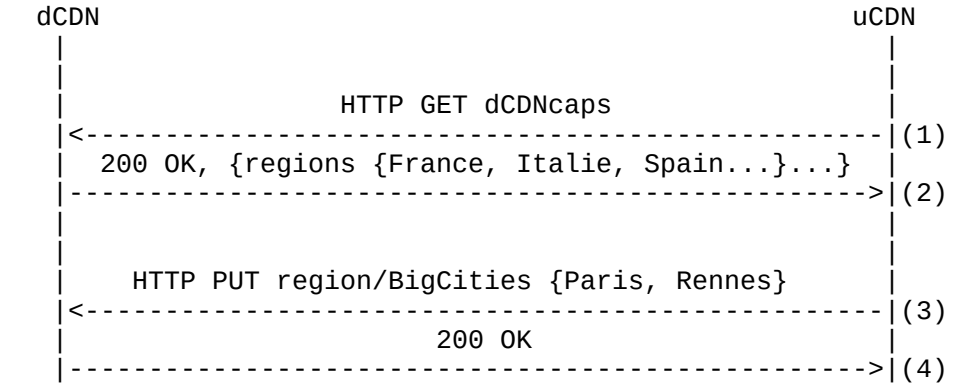


Figure 1, Initialization of the Mi interface

1. uCDN requests the details of the footprint where it is allowed to distribute contents;

2. dCDN returns the list of the countries;

3. uCDN creates a logical region named 'BigCities'. It initialize this region with 2 towns 'Paris' and 'Rennes' which are not geographically close;

4. dCDN accepts the creation;

3.2. Footprint Extension Use Cases

At first approach the CD metadata required to setup a footprint extension are intuitive. The uCDN simply indicates the geographic area to dCDN. Nevertheless there is a need of CDmD for controlling explicitly the delivery because dCDN is not aware of the constraints that apply to the contents.

3.2.1. Geographic Extension

in this use case uCDN interconnects with dCDN to extend its footprint to 2 countries covered by dCDN. Once the footprint extension setup achieved the RRi function of uCDN will be able to redirect the requests coming from the prefixes of these 2 coutries to dCDN RRi function.

Assumption:

the initialization of the Mi has be done previously as per the call flow of section 2.4: uCDN got the list of the countries of the footprint. dCDN initialized the main objects Italie and Spain.

Information elements:

Geographic capabilities are high level Geographic information like the name of well known areas, country, region, district, city...

Operations:

- Get the list of regions;

- Create a content;

- Initialize the contents to be distributed in a region;

- Create a region;

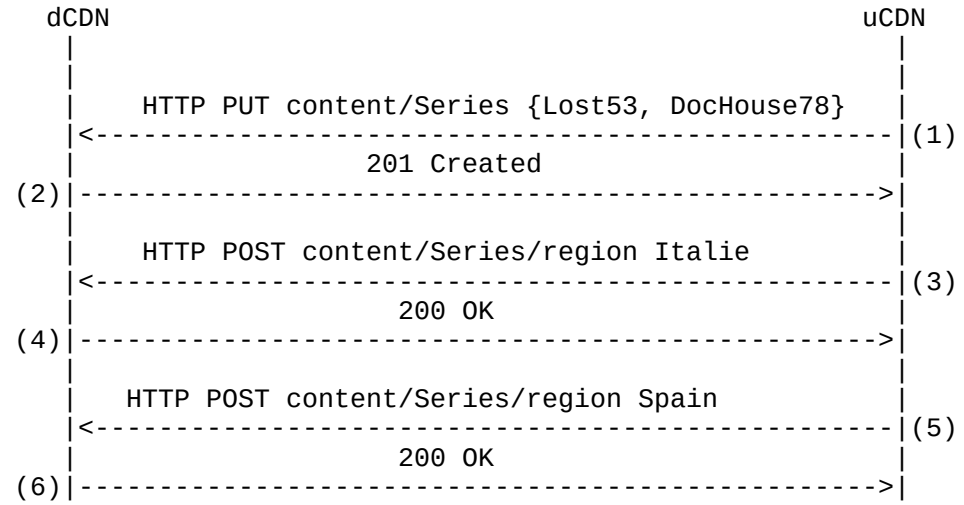


Figure 2, Geographic Extension

The initialization of the Mi is done according to the first call flow presented.

1. uCdn create the content 'Series' and initializes it with the pieces of contents 'Lost53' and 'DocHouse78'

2. dCDN accepts the creation;

3. uCDN adds this content in the region 'Italie' predefined by dCDN.

4. dCDN accepts the adding;

5. uCDN adds this content in the region 'Spain' predefined by dCDN.

6. dCDN accepts the adding;

Discussion:

uCDN might create a logical region 'South' initialized with 'Italy' and 'Spain' before step 1. This avoids steps 3 and 4.

Grouping CDmD may lead to complex processing and signaling when dCDN rejects the delivery of a subset of the contents.

3.2.2. Inter-Affiliates Interconnection

This use case covers the interconnection of CDNs managed by subsidiaries of a large group. For instance, it includes the interconnection of Orange France's and TP's CDNs, which are both part of the Orange group. The trust relationship among the interconnected CDNs is strong in this use case. Therefore, the CDNs can exchange internal dimensioning information like the number of caches, or detailed routing information, CDN detailed capabilities or performance information.

Information elements:

capacity { number of caches, sustainable sessions per second; sustainable delivery throughput...);

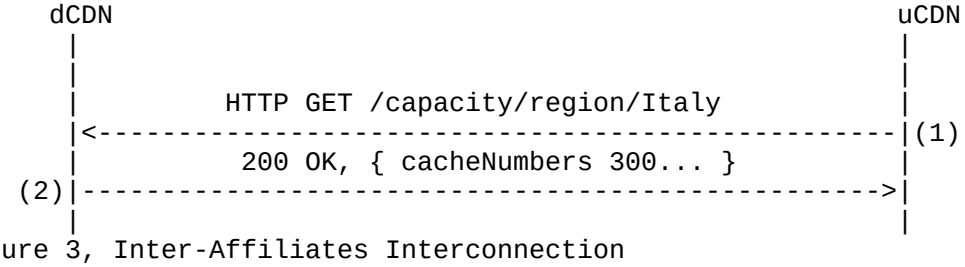


Figure 3, Inter-Affiliates Interconnection

1. uCDN requests the number of caches available on a region.

2. dCDN returns the information

Discussion:

Step1 gets directly the IE 'capacity' . In fact requesting the dCDNcaps at a whole may be enough to implement. This look like a XPATH request. Do we need a flag to present the level of granularity of the GET request ?

Such metadata exchanges enable tied interworking between the 2 CDNs.

At some extend, this kind of information may quickly overlap with monitoring information.

The object capacity must include a field 'extension' to permit enrichment.

3.2.3. On-Net Delivery

In this use case uCDN wants to deliver content to eyeballs directly connected to dCDN networks.

Information elements:

- on-net flag: indicate if a region include has an on-net footprint;

- on-net geoIP: the prefixes of network the dCDN eyeballs;

Operation:

- Get on_net regions;

- get on_net region prefixes;

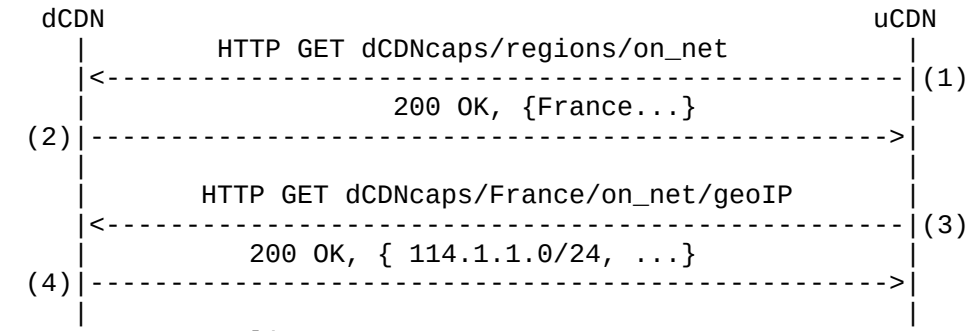


Figure 4, On-Net Delivery

1-2. uCDN downloads the regions to which the dCDN can deliver on-net. This information enables the uCDN to know which areas are served.

3-4. uCDN gets the prefixes. Then it adapts its CDN selection policies to guarantee that the content will always be delivered on net, with the best performance (i.e. uCDN redirects to dCDN only the content requests coming from eyeball located on dCDN networks) .

3.2.4. Nomadic Users

This use case considers that the initialisation of the Mi has been done before as per section 3.1.

Asumption:

For optimization reasons uCDN and dCDN did not provision the distribution of the content by dCDN. Consequently the CDmD of the content and the content are exchanged on-the-fly between the 2 CDNs and the content is purged at the end of the delivery.

information element:

- synchonous purge: a flag which indicate that the purge must be done at the end of the delivery after a reasonable (from cache algo point of view) delay
- a timer of a a reasonable delay;
- on the fly acquisition: flag indicating that dCDN supports or not on the fly content acquisition;

Operations:

- Push of per content metadata in dCDN;

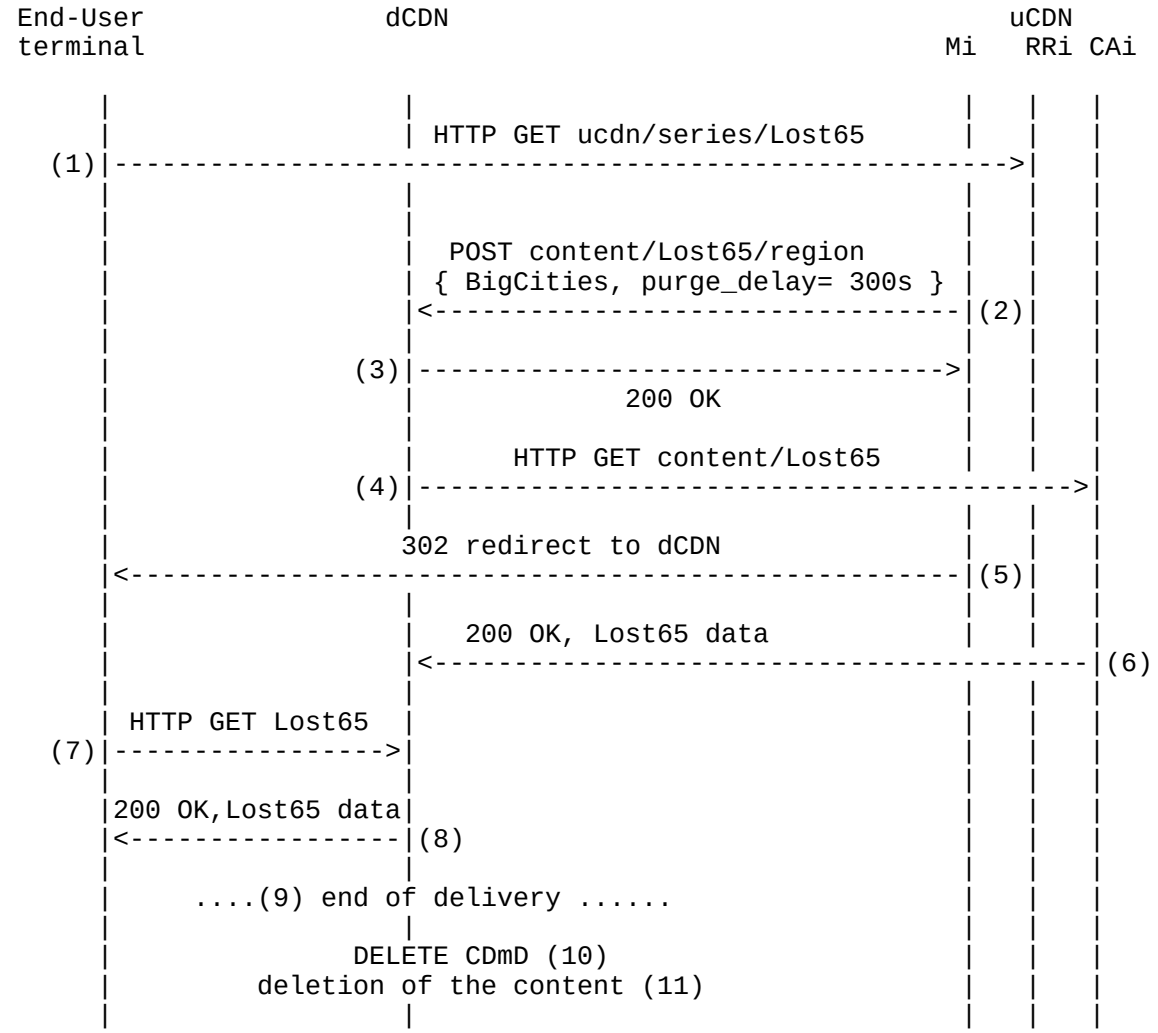


Figure 5, On the fly CDmD exchange for Nomadic use case

1. A end-user requests a content that uCDN is in charge of.

2. The request arrives on the request routing function of uCDN; The request routing logic of uCDN determines that the end-user is located on the footpring of dCDN; dCDN pushes the CDmD of this content toward dCDN. CDmD includes information describing the constraint attached to the nomadic case (limited to one user, ...)

3. dCDN accepts the delivery.

4. dCDN starts the acquisition of the content.

5. UCDN redirects the request to dCDN.

6. CDN stores the (first part) of the content

7. The end-user request is received by dCDN.

8. dCDN starts the distribution of the content to the end-user.

9. The delivery achieved;

10. dCDN deletes of the CDmD

11. dCDN delete the content

Discussion:

In this use case only the user considered in (5) must be served by dCDN. this does not means that dCDN must check the identity of the user in (7) because if any identity verification must be performed it should be checked by uCDN in (5) before the redirection.

Most of the use cases require an Mi operation to explicitly delete the metadata attached to a content;

3.3. Offload Use Cases

This section gives examples of call flow for the offload use cases.

3.3.1. Overload Handling and Dimensioning

The initialization of the Mi has been done in section 3.1.

The CDN interconnection is setup to cover unexpected peak of traffic in uCDN.

Information element:

- Content object;

- Capacity:peak (unit, value), sustainable: (unit, value); duration (start, end);

- bursty interconnection flag: a flag to clearly distinguish very bursty interconnection from more stable ones.

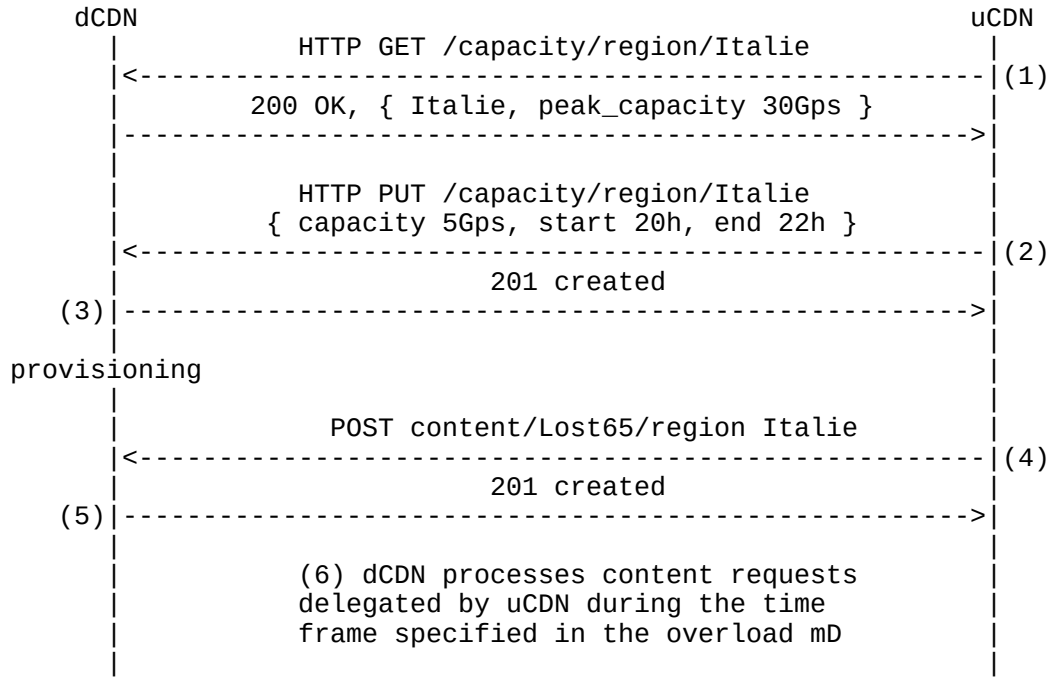


Figure 6, Overload handling, planned traffic peak

1. uCDN downloads dCDN overload traffic handling capabilities (may be downloaded during bootstrapping). This information enables the uCDN to know how much traffic it can delegate to the dCDN.

2. uCDN creates capacity request mD. This enables the dCDN to know how much traffic and the period of traffic the uCDN will delegate to the dCDN and (e.g., for planned traffic peaks, or planned offload for maintenance operations).

3. dCDN acknowledges that it understands and agrees to the overload mD. Then it provisions the ressources.

4. uCDN creates geographic content delivery restrictions CDmD pushing the CSP related policies to dCDN. This enables the dCDN to know to which areas it must or must not deliver every piece of content.

5. dCDN acknowledges that it understands and accpepts the CDmD.

Notes: if the dCDN cannot or does not want to fulfill the capacity request, it will response with an HTTP error message such as a 403 forbidden. This use case covers the failure of delivery resources. It assumes that the uCDN is still able to redirect requests to the

dCDN, but not to serve all the requests by itself.

Discussion:

This use case highlights the information elements of a regular CDNi interconnection (even if peak and sustainable value are extremely different in an Offload situation).

3.3.2. Resiliency

3.3.2.1. Failure of Content Delivery Resources

uCDN interconnects with dCDN1 and dCDN2 to guarantee service continuity when dCDN1 can not handle the delivery.

The call flow below presents the situation where dCDN1 encounters an overload problem or a failure before beginning the delivery and cannot serve the content to the UE.

Assumptions:

dCDN1 and dCDN2 have similar footprint. uCDN already pushed the CDmD of the content named Lost65 both on dCDN1 and dCDN2. n normal situation dCDN1 distributes the content 'Lost65' and already made the acquisition of the content Lost65.

Information elements:

- on_dCDN_failure_FQDN: FQDN of the uCDN RRi which handles the redirection on failure of the redirection function of dCDN.

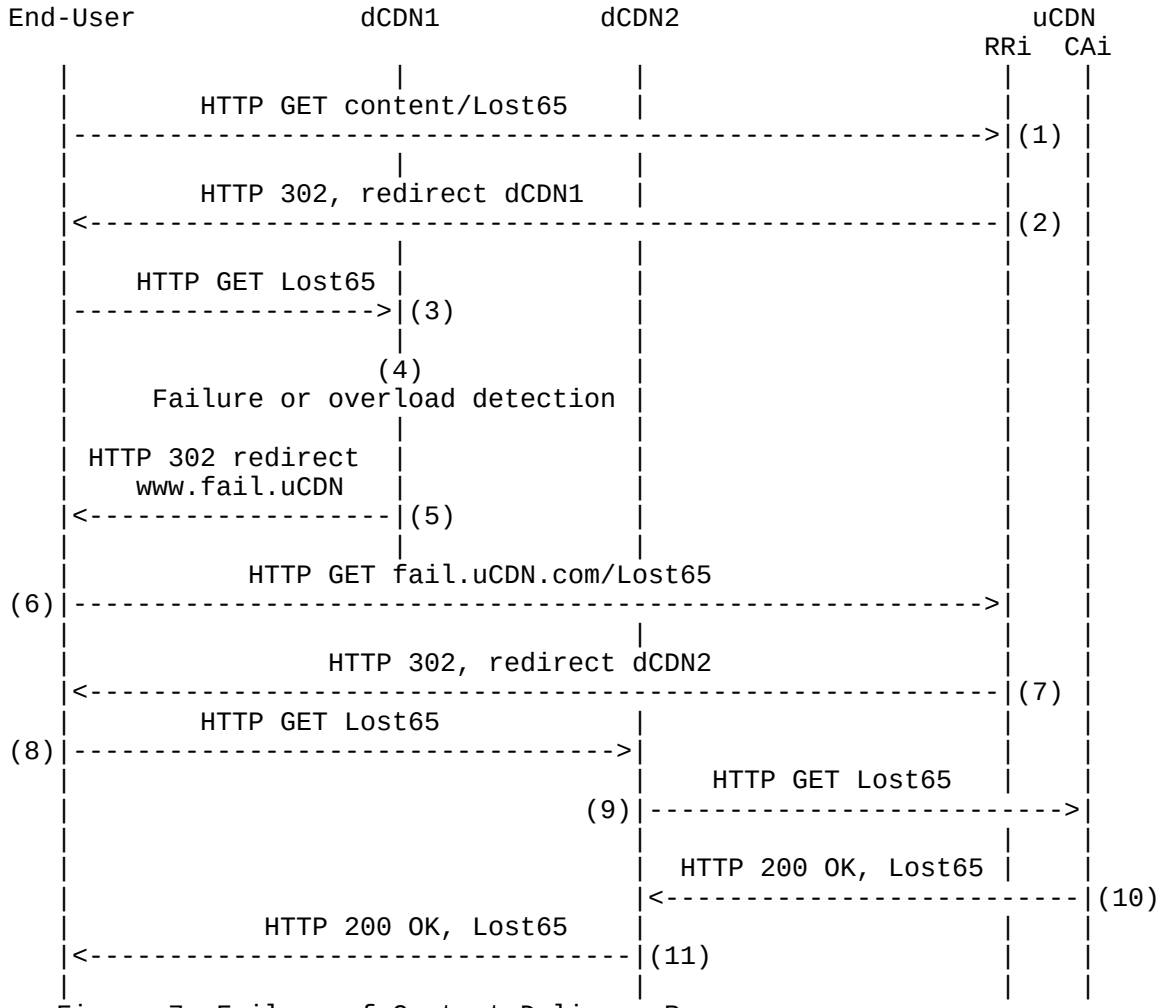


Figure 7, Failure of Content Delivery Resources

1. A end-user requests a content that uCDN is in charge of;

2. uCDN redirects the request to dCDN1;

3. The end-user request is received by dCDN1;

4. dCDN detects a failure or a lack of ressource on its delivery;

5. dCDN1 redirects the EU request to a dedicaced FQDN RRi of uCDN which handles the failure of delivery;

6. The end-user request is received by uCDN on this special FQDN

(Editor notes: solution already presented in another draft: include the reference);

7. uCDN redirects the request to dCDN2;

8. The end-user request is received by dCDN2;

9. dCDN2 starts the acquisition of the content;

10. dCDN2 stores the content;

11. dCDN2 sends the content to the EU;

Discussion:

Direct redirection between dCDN1 and dCDN2 may reduce the redirection duration. it requires the exchange of more CDmD and hides the failure of the delivery to uCDN.

In this use case the detection of the failure happens before the selection of the delivery node. In case of failure during the delivery, dCDN should send a message to uCDN on the control interface.

3.3.2.2. Failure of Content Acquisition

Assumption:

uCDN interconnects with dCDN1 and dCDN2 for the delivery of the content 'Lost65'.

dCDN2 already made its acquisition and keep a copy.

uCDN setups dCDN2 as a backup solution for the acquisition of 'Lost65'.

Information element :

- backupCDN: FQDN of a content acquisition backup says that the upstream CDN provides (or not) a backup for the acquisition of the content it is requesting the distribution.

- storage_duration: duration of the storage of a content.Flag of the content object to request that the dCDN keep a copy of the content during a period of time after acquisition (or after the last delivery);

- immediat_acquisition: flag requesting that dCDN must start the acquisition triggered on reception of the content metadata;

operations:

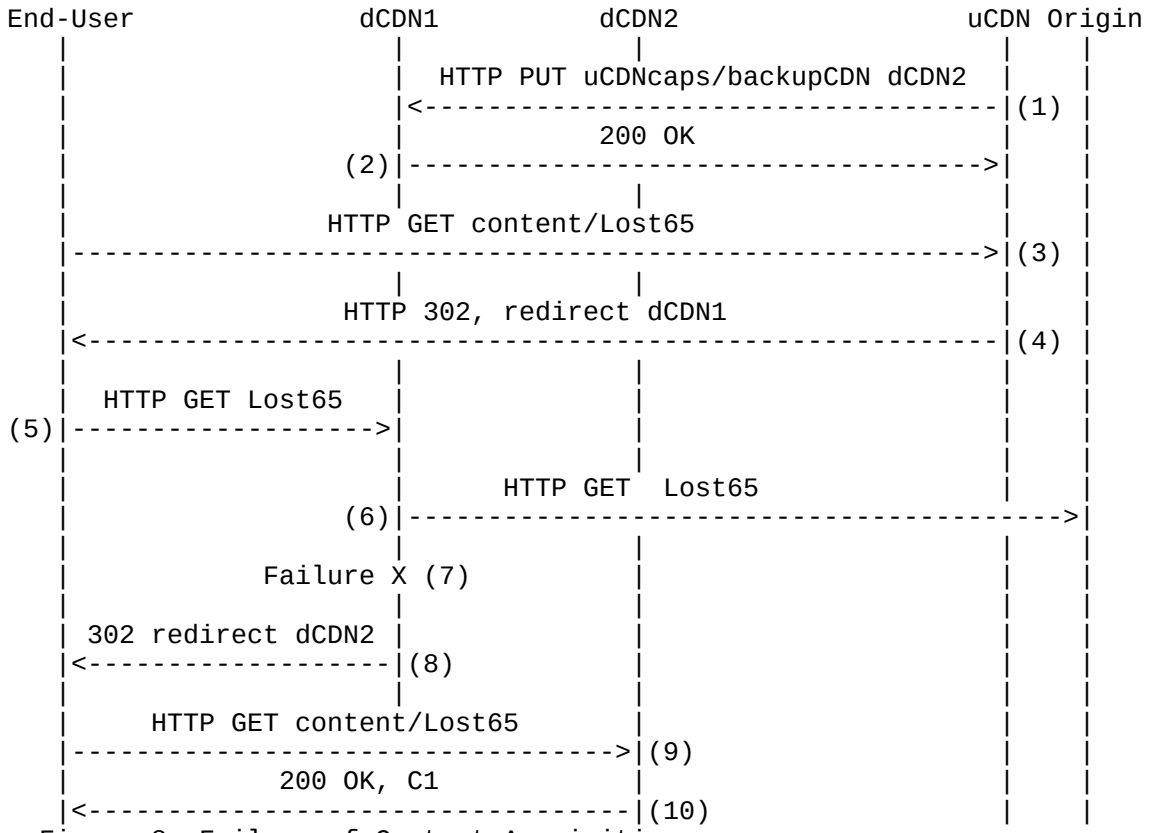


Figure 8, Failure of Content Acquisition

1.uCDN setups the backup CDN.

2. dCDN1 accepts.

3. End-user requests the content Lost65.

4. uCDN redirects the request to dCDN1.

5. The end-user request is received by dCDN1

6. dCDN1 starts the acquisition of the content.

7. The acquisition fails

8. dCDN1 redirects the UE to dCDN2 because uCDN provided a backup.

9. The end-user request is received by dCDN2.

10. dCDN2 deliver the content to the UE.

Discussion:

The failure of content acquisition may be solved without redirection, simply with the selection of a backup acquisition server.

3.4. CDN Capability Use Cases

3.4.1. Device and Network Technology Extension

Assumptions:

uCDN only streams content using HTTP smooth streaming protocol. dCDN supports other protocols like MPEG-DASH. An End-user requests a MPEG-DASH content that is managed by uCDN. The uCDN and the dCDN have exchanged their capabilities prior the UE content request. uCDN has pushed content related metadata before the EU request.

Information elements:

- delivery methods supported (HTTP smooth streaming, MPEG-DASH,...);

- networks type supported (fiber, xDSL, WiFi, 3G, LTE,...);

The figure below presents the typical CD mD call flow:

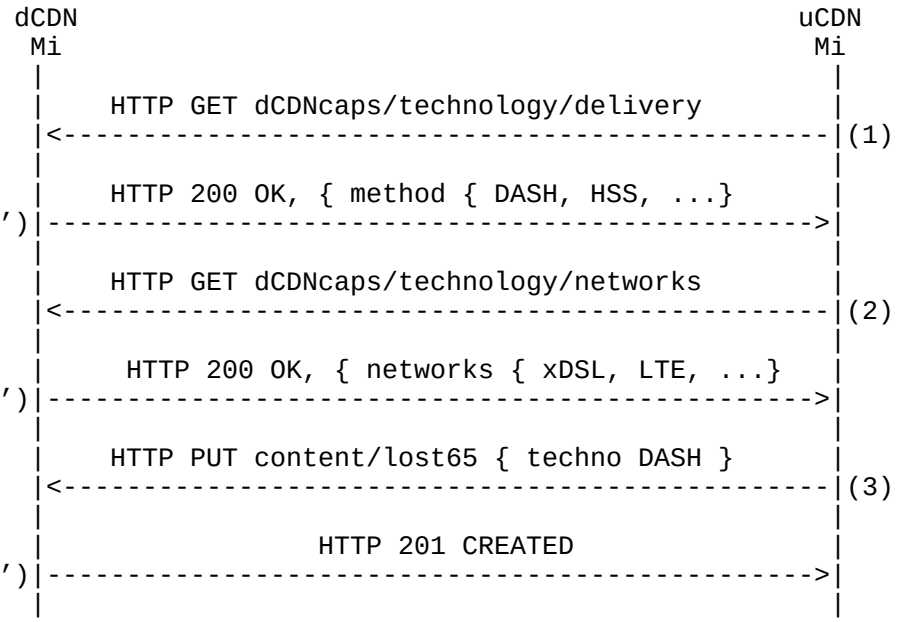


Figure 9,Device and Network Technology Extension

1. uCDN downloads dCDN delivery technology capabilities.

2. uCDN downloads dCDN network type capabilities.

3. uCDN pushes the CDmD of this content toward dCDN. CDmD includes information describing the constraint attached to the CDN capability use case.

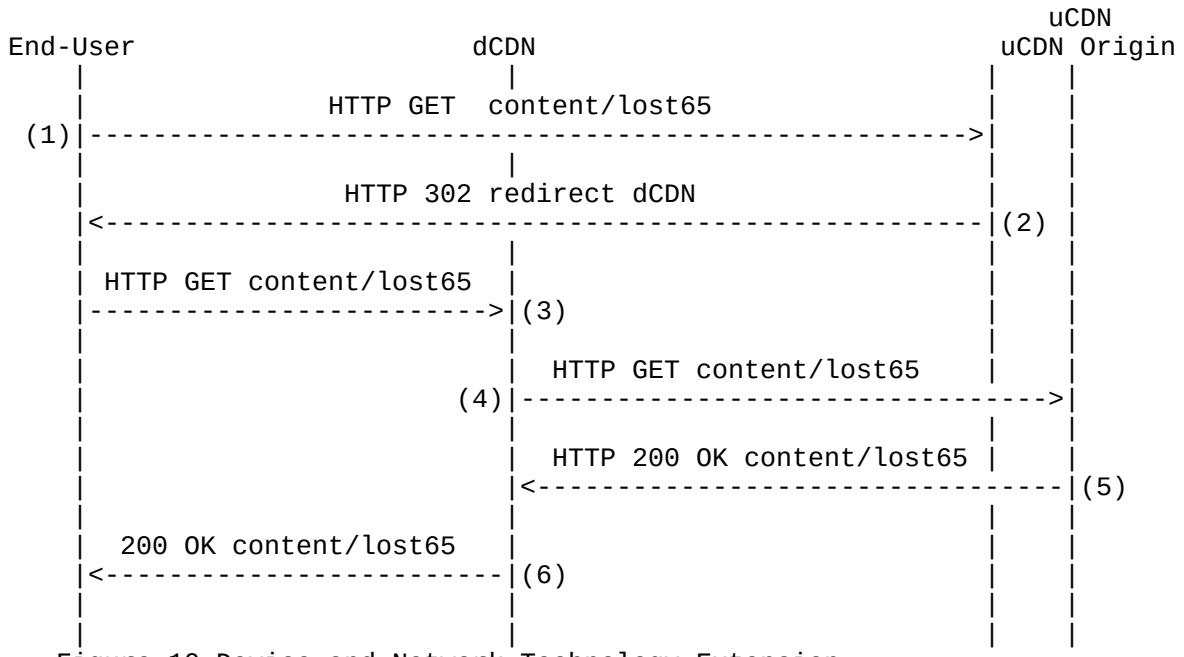


Figure 10,Device and Network Technology Extension

call flow:

1. A end-user requests a content that uCDN is in charge of.

2. uCDN redirects the request to dCDN.

3. The User Equipment request is received by dCDN.

4. dCDN starts the acquisition of the content.

5. CDN stores the content

6. The delivery achieved

Discussion :

Synchronous vs asynchronous : The exchange of the capabilities between the CDN may be done before receiving a content request or when receiving a content request. This case adds delay to the handling of the content request. The content related metadata may be provisioned before the request.

3.4.2. Technology and Vendor Interoperability

In a CDN interconnection, CDN may need to exchange their configuration : CDN parameters, functions configuration. Such information can be provided at service bootstrapping, but can also be provided on fly using the metadata interface.

information elements:

Description of a CDN:

- origin server
  - orgin servers list (server name, IP)
  - ingestion interface: IP addresse of the ingestion interface of origin server
- routing server
  - list: routing server name lists
  - routing interface ip: provides the routing interface IP of the routing server name
- token
  - format: hash algorithm used
  - parameters: parameters URL used to calculate token (domain, keys, values)
  - key: used to calculate token
- url
  - format: format of the URL
  - parameters: name of the parameters
  - function: name of the function (represented by parameter). Such function should be understood by the uCDN.

Call flow:

In this use case, we assume that uCDN wants to get the URL and token

format in order to be able to check URL and token and adapt them if necessary when redirecting end-users to dCDN.

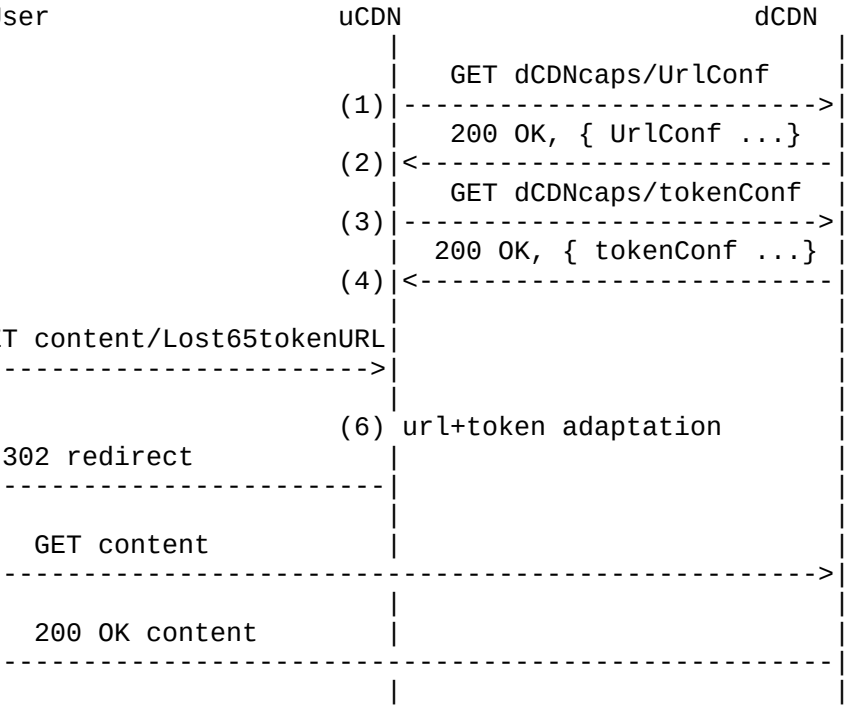


Figure 11, Technology and Vendor Interoperability

Call flow:

1. uCDN requests the dCDN URL configuration metadata;

2. dCDN answers with its URL configuration metadata;

3 . uCDN requests the dCDN token configuration metadata;

4. dCDN answers with the token configuration metadata (e.g. CDN model hash algorithm = MD5, ciphering key = a1f4d0eab4df...)

5. A end-user requests a content that uCDN is in charge of.

6. According to information received, the uCDN adapts its policies : redirection, ingestion

7. uCDN redirects the request to dCDN.

8. End-user requests the dCDN to deliver content

9. dCDN accepts the delivery.

Discussion:

Security considerations:

Most of this information can be restricted to specific CDNi members and subject to access control rights. Thus members SHALL be authentified before accessing those data. Requested CDN (uCDN or dCDN) MAY refuse access to information by issuing a 401unauthorized response.

3.4.3. Dynamic QoE and QoS improvement

In this use case, the uCDN will check if the delivery QoE of the dCDN is compliant with QoE of the CDNi SLA. If not compliant, then the uCDN will redirect next requests to other dCDNs.

Assumption:

uCDN indicates the SLA that dCDN must ensure high level QoE (no sessions failures, or glitches at client side).

information elements:

- Policy: Policy provides CDN policy to ensure QoE (typically can tell which specific function can be ensured by the CDN)

- QoE: Key Performance indicator assessing the QoE (could be computed for instance on% gliches, % sessions failures, % access to service delay, etc.).

call flow:

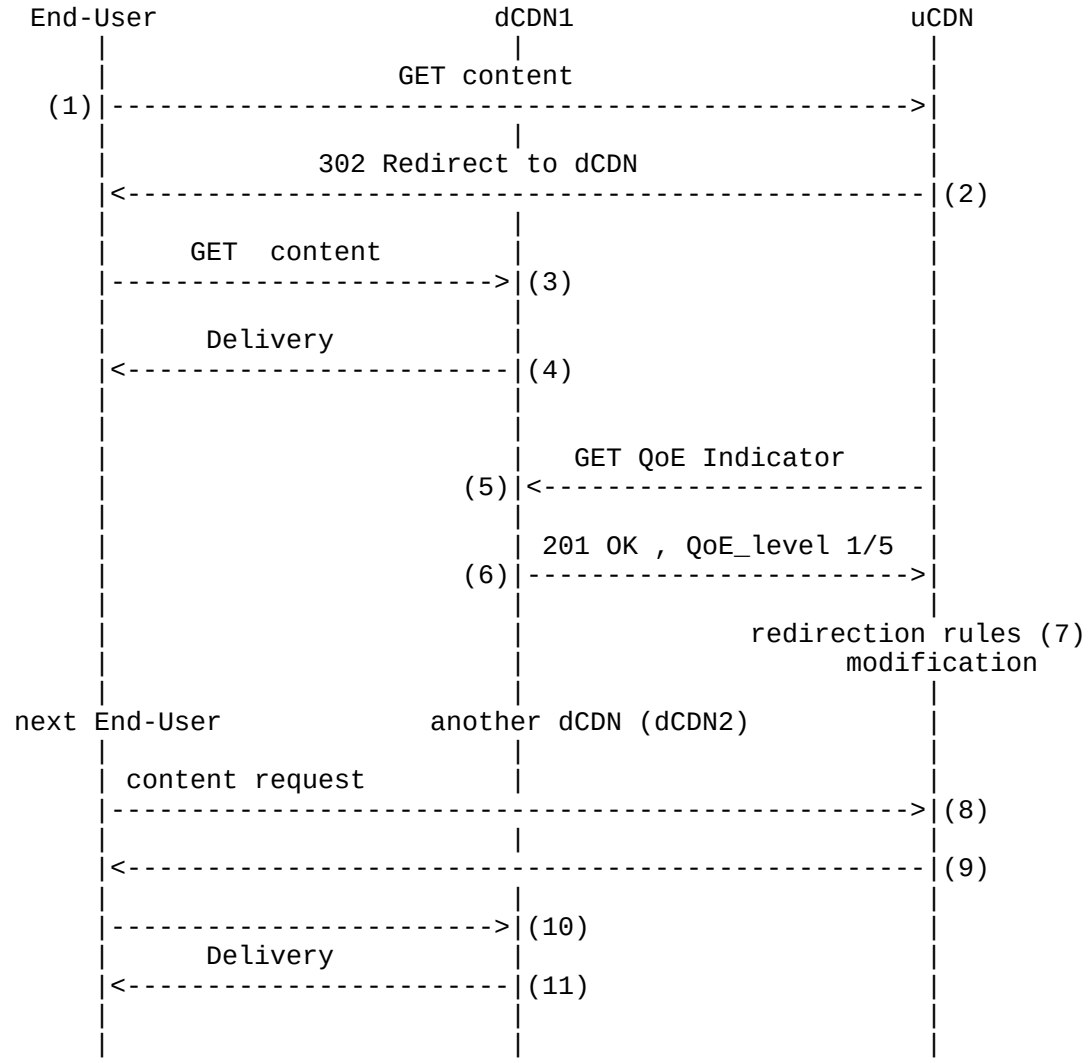


Figure 12, QoE and QoS improvement

1. A end-user requests a content that uCDN is in charge of.

2. uCDN redirects the request to dCDN1.

3. End-user requests dCDN1 to deliver content

4. dCDN1 accepts the delivery.

5. uCDN requests the QoE indicator from dCDN1 for ongoing delivery

sessions.

6. dCDN1 sends the QoE level indicator.

9. uCDN adapts its redirection rules.

10. Another end-user requests a content that uCDN is in charge of.

11. uCDN redirects the request to dCDN2 with the initial QoE.

12. End-user requests the dCDN2 to deliver content

13. dCDN2 accepts the delivery.

Discussion:

Retrieving QoE information may need some adaptation in the player at client side.

Statistics data imply logs data processing at CDN side. Statistics are carried over a the monitoring interface, and therefore are not metadata. There computing takes time and may delay the detection of the decrease of QoE.

4. Discussions

This section gather points to present during the meeting or to discuss on the ML.

4.1. JSON reference

What is the reference of the JSON language ? is it only [RFC4627] ?

Is there a JSON framework for specifying things like XML Schema ?

4.2. Network and infrastructure Metadata

2 CDNs may desire to exchange information on the location, the routing, the reachability, the availability and the load of their ressources. These metadata are not content distribution metadata.

5. Inputs for the next version

5.1. Requirement

Add the link toward individual entries of the requirement draft at the format [Req #].

Identify and specify new requirements for [I-D.ietf-cdni-requirements].

6. IANA Considerations

None by now.

7. Security Considerations

This section needs more works:

Content distribution Metadata, include information that may ease DoS towards CSP or CDN infrastructures.

Privacy: Content distribution Metadata may carry information on the location of the terminal

8. Acknowledgements

The authors would like to thank Yannick Le Louedec for its reviews and comments.

Part of this work is funded by the EU FP7 Ocean project.

9. References

9.1. Normative References

[I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-00 (work in progress), September 2011.

[I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-01 (work in progress), October 2011.

[I-D.ietf-cdni-use-cases]
Bertrand, G., Emile, S., Watson, G., Burbridge, T., Eardley, P., and K. Ma, "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-00 (work in progress), September 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

9.2. Informative References

[ATIS-0800042]
ATIS, "ATIS IPTV Content on Demand Service", 2010, <ATIS IPTV Content on Demand Service>.

[DVB-IP-TV]
Digital Video Broadcasting (DVB), "Transport of MPEG-2 TS Based DVB Services over IP Based Networks", 2003, <http://www.etsi.org/deliver/etsi_ts/102000_102099/102034/01.04.01_60/ts_102034v010401p.pdf>.

[ETSI-3GP-DASH-R10]
ETSI, "Progressive Download and Dynamic Adaptive Streaming over HTTP", 2010, <http://www.3gpp.org/ftp/Specs/html-info/26247.htm>.

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

[I-D.jenkins-cdni-metadata]
Niven-Jenkins, B., Ferguson, D., and G. Watson, "CDN Interconnect Metadata", draft-jenkins-cdni-metadata-00 (work in progress), September 2011.

[I-D.ma-cdni-metadata]
Ma, K., "Content Distribution Network Interconnection (CDNI) Metadata Interface", draft-ma-cdni-metadata-00 (work in progress), October 2011.

[I-D.pantos-http-live-streaming]
Pantos, R., "HTTP Live Streaming", draft-pantos-http-live-streaming-07 (work in progress), September 2011.

[I-D.peterson-cdni-strawman]
Peterson, L. and J. Hartman, "A Simple Approach to CDN Interconnection", draft-peterson-cdni-strawman-01 (work in progress), May 2011.

[I-D.stiemerling-cdni-routing-cons]
Stiemerling, M., "Considerations on Request Routing for CDNI", draft-stiemerling-cdni-routing-cons-00 (work in progress), July 2011.

[I-D.thompson-cdni-atis-scenarios]
Thompson, B. and A. Kobayashi, "ATIS Internet Sourced Content Initiative and Relevance to CDNI", draft-thompson-cdni-atis-scenarios-00 (work in progress), March 2011.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[SNIA-CDMI-1.0]
ATIS, "Cloud Data Management Interface", 2010, <http://www.xmlmind.com/xmleditor/namespace/clipboard>.

[TVA-CD-mD]
Mtv-anytime, "Metadata Specification Version 1.3", 2003, <http://www.tv-anytime.org/workinggroups/wg-md.html>.

Appendix A. GPX XML Schema fields exensibility

Following is an example of extension in XML. GPX 1.1 (See http://www.topografix.com/GPX/1/1/) is a popular datamodel for geolocalization information exchange. GPX includes GPS localization (way point) and navigation information (routes and tracks). Each object includes an ’extensions’ element.

localization information elements

```
<xsd:complexType name="gpxType">
<xsd:sequence>
<xsd:element name="metadata" type="metadataType"/>
<xsd:element name="wpt" type="wptType"/>
<xsd:element name="rte" type="rteType"/>
<xsd:element name="trk" type="trkType"/>
<xsd:element name="extensions" type="extensionsType" />
</xsd:sequence>
</xsd:complexType>
```

Metadata information elements

```
<xsd:complexType name="metadataType">
<xsd:sequence>
<-- elements must appear in this order -->
<xsd:element name="name" type="xsd:string"/>
<xsd:element name="desc" type="xsd:string""/>
<xsd:element name="author" type="personType"/>
<xsd:element name="copyright" type="copyrightType"/>
<xsd:element name="link" type="linkType"/>
<xsd:element name="time" type="xsd:dateTime"/>
<xsd:element name="keywords" type="xsd:string"/>
<xsd:element name="bounds" type="boundsType"/>
<xsd:element name="extensions" type="extensionsType"/>
</xsd:sequence>
</xsd:complexType>
```

Figure #, GPX XML Schema Extension field

Authors' Addresses

Stephan Emile
France Telecom Orange
2 avenue Pierre Marzin
Lannion F-22307
France

Email: emile.stephan@orange.com

Bertrand Gilles
France Telecom Orange
38-40 rue du General Leclerc
Issy Les Moulineaux F-92794
France

Email: gilles.bertrand@orange.com

Fieau Frederic
France Telecom Orange
2 avenue Pierre Marzin
Lannion F-22307
France

Email: fieau.frederic@orange.com

Pages Remi
France Telecom Orange
38-40 rue du General Leclerc
Issy Les Moulineaux F-92794
France

Email: remi.pages@orange.com

CDNI Working Group
Internet Draft
Intended status: Standards Track
Expires: April 2012

Xiaoyan.He
Jincheng.Li
Spencer.Dawkins
Huawei
Ge.Chen
China Telecom
October 13, 2011

Request Routing Protocol for CDN Interconnection
draft-xiaoyan-cdni-request-routing-protocol-00.txt

Status of this Memo

Copyright Notice

Abstract

The Request Routing Protocol allows the Request Routing system in interconnected Content Delivery Network(CDNs) to communicate to ensure that an end user request can be (re)directed from an upstream CDN to a surrogate in the downstream CDN. This document describes the details of the protocol used to provide this mechanism.

Table of Contents

1. Introduction

A Content Delivery Network(CDN) is a system of computers built on an existing IP network which is used for large scale content delivery, via prefetch or cache contents to its distributed computers close to the end users, a CDN can improve access to the data it caches, reduce access latency and improve end user's experience.

In recent years the volume of video and multimedia content delivered over the internet is rapidly increasing. To accommodating this increase, existing CDN providers are scaling up their infrastructure and many Network Service Providers (NSPs) are deploying their own CDNs.
Another emerging requirement is CDN Interconnection (CDNI).
Several real world use cases are described in [I-D.draft-cdni-use-cases] to prove the necessity for CDN interconnection. The most frenquently mentioned use case is via leveraging the collective CDN footprint of interconnected standalone CDNs to achieve the goal of delivering content to additional distributed end users regardless of their location.

As there is no existing standard to facilitate CDN interconnection, IETF has established a working group to produce specifications needed. This draft is written in response to the problem area described in [I-D.draft-cdni-problem-statement], when CDNs are interconnected as described in [I-D.draft-cdni-use-cases] based on the requirements described in [I-D.draft-cdni-requirements], and using the technology framework described in [I-D.davie-cdni-framework].

The purpose of this document is to define the request routing protocol for CDNI, which is one of the main building blocks of the CDN interconnection architecture described in [I-D.draft-cdni-requirements].

1.1. Terminology

This document reuses the terminology defined in [I-D.draft-cdni-problem-statement].

The term "Distinguished CDN Domain" defined in [I-D.davie-cdni-framework] also reused in this document.

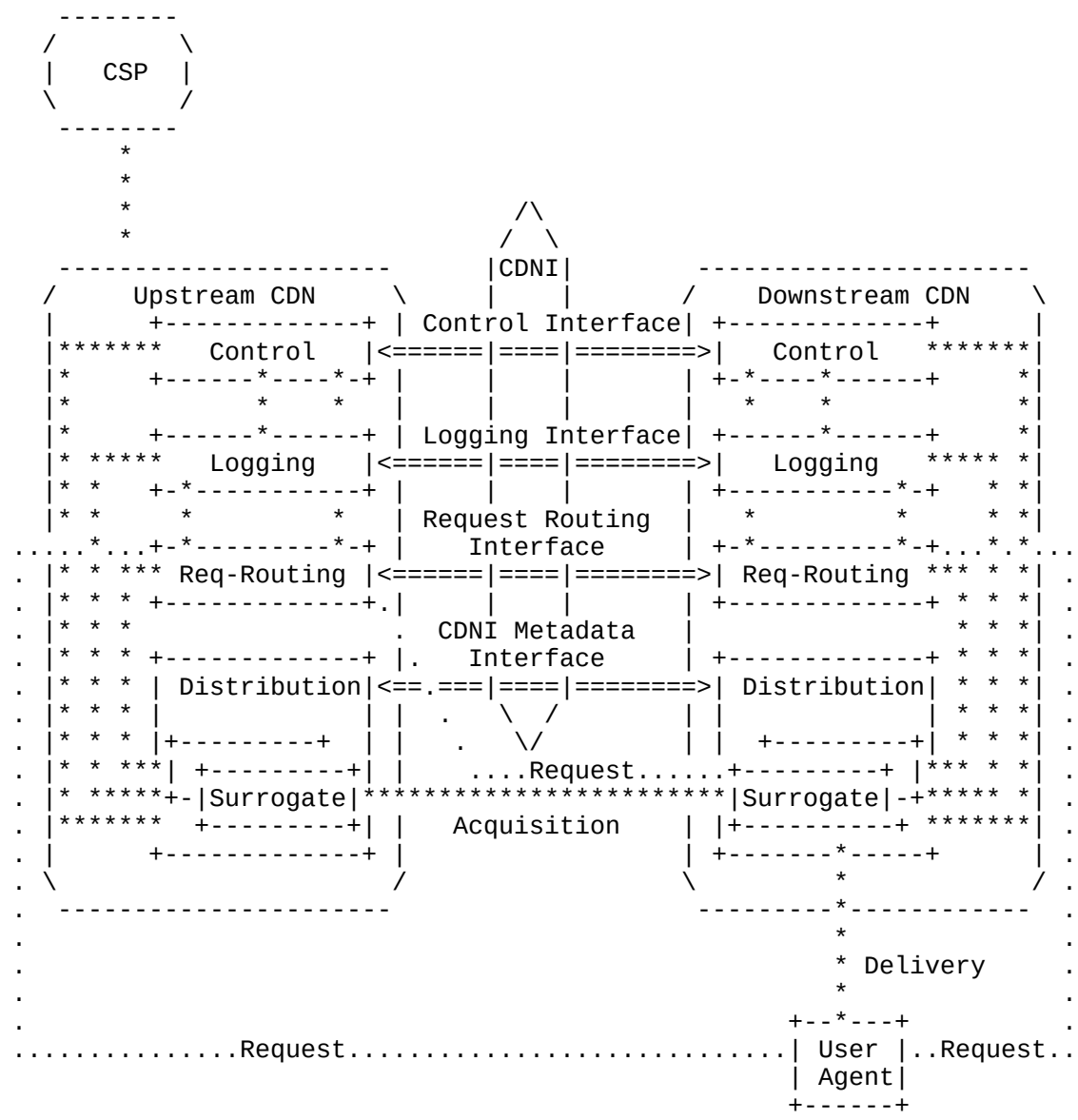
The following terms are also used by this document:

DNS Redirection: The act of using DNS name resolution for the routing process of a CDN. In DNS Redirection, the DNS resolver of the CDN makes the routing decision based on a local policy and returns the result as the response of a DNS query request to redirect a user agent to a new target. In CDNI, the result may point to a surrogate of the CDN, another interconnected CDN etc.

HTTP Redirection: The act of using an HTTP redirection response to redirect a user agent to a new target. The new target is the result of the routing decision of a CDN at the time it receives a content request via HTTP. In CDNI, the result may point to a surrogate of the CDN, another interconnected CDN. etc.

1.2. Reference Model

Figure 1 from [I-D.draft-cdni-problem-statement] illustrating the CDNI model and the CDNI protocols is replicated below. This document describes the Request Routing Protocol shown in the figure.



<==> interfaces inside the scope of CDNI
**** interfaces outside the scope of CDNI
.... interfaces outside the scope of CDNI

Figure 1: CDNI Model

2. Conventions used in this document

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].

3. Protocol Function and Operation Overview

The Request Routing Protocol is one of the main building blocks for CDNI. The main function of the Request Routing Protocol is to allow the Request-Routing systems (see Figure 1) in interconnected CDNs to communicate to facilitate redirection of the request across CDNs. In particular, its function can be summarized as follows:

* allow the Upstream CDN (uCDN)to query the Downstream CDN (dCDN) at request-routing time before redirecting the request to the Downstream CDN.
* allow the Downstream CDN to provide to the Upstream CDN (static or dynamic) information (e.g. resources, footprint, load) to facilitate selection of the Downstream CDN by the Upstream CDN request routing system when processing subsequent content requests from User Agents.

The detailed requirements which the Request Routing Protocol need to meet and priorities of those requirements are described in section 5, [I-D.draft-cdni-requirements].

To enable the communications over the Request Routing Interface, the two interconnected CDNs need to know each other’s contact address(es). For example, an Upstream CDN needs to know the contact address of a Downstream CDN to send a query request based on HTTP for redirection preference, or a Downstream CDN needs to know the contact address(es) of the upstream peer it should report its capability information to.

The contact address may be statically pre-configured, dynamically discovered via control interface, or other means. However, they are not specified in this document, as this is considered not in the scope of the CDNI Request Routing Protocol.

3.1. Request Routing

The CDNI solution must support two request routing mechanisms. As illustrated in section 3.2 and 3.4 of [I-D.davie-cdni-framework], the Iterative Request Routing method does not invoke any interaction over the request routing interface across interconnected CDNS. This document will not discuss Iterative Request Routing further.

In the case of Recursive Request Routing, an Upstream CDN forwards a routing request from a user agent to a Downstream CDN for surrogate selection. The candidate protocols for these interactions are DNS and HTTP. Moreover, the routing mechanisms used between the CDN and the user agent (DNS and HTTP Redirection) of the two interconnected CDNs should also be taken into account as they may affect the type of query request the Upstream CDN send to a Downstream CDN and the information the Downstream CDN may send in its query response.

The request routing procedure has several variants depending on the factors including:

O Which routing mechanism is adopted by an Upstream CDN, DNS Redirection or HTTP Redirection.

O Which protocol is adopted over the Request Routing Interface, DNS or HTTP.

O Which routing mechanism is adopted by a Downstream CDN, DNS Redirection or HTTP Redirection.

All possible combinations and their validity are shown in Table 1.

CaseNO.	uCDN Received Request	RRI Interface	dCDN Response	Note
1	DNS	DNS based	DNS with IP address of RR	dCDN works in HTTP Redirection mode, illustrated in section 3.1.1.1.
2	DNS	DNS based	DNS with hostname of RR	dCDN works in DNS Redirection mode, illustrated in section 3.1.1.2.
3	DNS	HTTP based	Invalid case	Protocol conversion occurs in uCDN, invalid case.
4	HTTP	HTTP based	HTTP 302 Redirection	dCDN works in HTTP Redirection mode, illustrated in section 3.1.2.
5	HTTP	HTTP based	DNS Redirection	dCDN works in DNS Redirection mode, invalid case.
6	HTTP	DNS based	Invalid case	Protocol conversion occurs, invalid case.

Table 1: Recursive Routing Cases

The rules to filter the cases and determine the validity of them are discussed below.

The Upstream CDN must not perform protocol conversion (A DNS query to an HTTP request or vice versa). To assist the routing decision of a Downstream CDN, the Upstream CDN conveys as much information as possible to the Downstream CDN, e.g. URI of the requested content, the client’s location information. In the case of HTTP to DNS conversion, a DNS request cannot convey all the information an HTTP request contains. In the case of DNS to HTTP conversion, a full HTTP URL cannot be constructed through a simple

domain name contained by a DNS query request. Hence it is concluded that the protocol type used in the Request Routing Interface will be consistent with the one the Upstream CDN received from the user agent. Case3, Case6 are invalid according to this rule.

The Downstream CDN can determine according to its local policy a DNS Redirection or an HTTP Redirection to be adopted. When receiving a DNS query request over the Request Routing Interface. If DNS Redirection is selected, as the location information has been changed to the Upstream CDN's when it proies the DNS query request, the Downstream CDN cannot get the user agent's location information from the query request. The Downstream CDN sends a response with a CNAME with the hostname of the Request Router, so that the user agent issues another DNS query request which will convey its location information as shown in case2. If HTTP Redirection is selected, the Downstream CDN sends a response with the IP address of its Request Router, so that it can receive a subsequent content request based on HTTP containing the client's location information, to allow selection of an appropriate surrogate as shown in case1.

Based on filter rules above, Case 1, 2, and 4 are valid cases for CDNI. The following section describes these cases in detail.

3.1.1. DNS based Request Routing Protocol

3.1.1.1. HTTP Redirection utilized by Downstream CDN

This example illustrates the CaseNo1 of Table 1. Based on local policy, the Upstream CDN adopts the DNS Redirection with the user agent while the Downstream CDN utilizes the HTTP Redirection. The Downstream CDN should return the IP address in an RR. In this example, the distinguished domain name of the Downstream CDN is "cdni.op-b.example".

Note: To simplify the presentation, the full URL of the HTTP GET message is not shown in the example figures of this document. Only the FQDN at the beginning of each URL is explicitly presented, however the rest of the URL e.g. the path

parameters contained in the URL should be considered to be present.

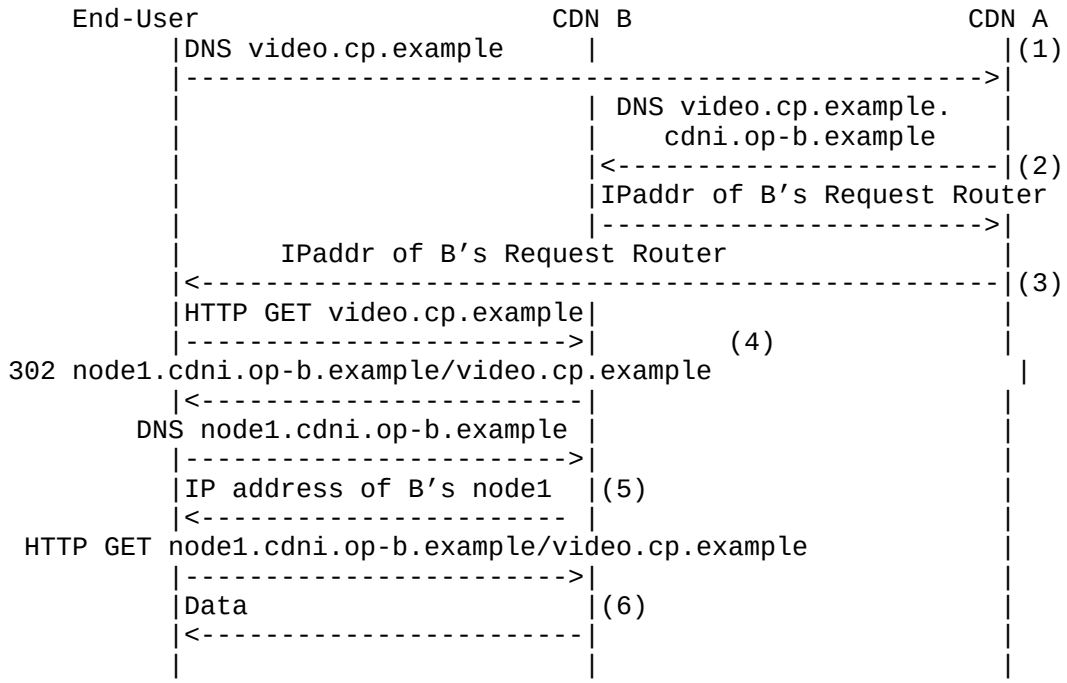


Figure 2 DNS based CDNI Recursive Request Routing 1

1. A Request Routing System of CDN A processes the DNS request for its customer based on the domain video.cp.example and recognizes that the end-user is best served by another CDN, specifically CDN B. Based on the pre-configured distinguished domain name of CDN B and a negotiated rules for constructing a domain name contained in a DNS query request over RRI that have been negotiated with CDN B, the Request Routing System changes the domain name to CDN B's domain name, with the CP's domain information, and acts as a proxy server forwarding the DNS request to CDN B.

Note: For simplicity, the local DNS invoked in the procedure is not shown.

2. Based on the local policy, CDN B determines that the routing mechanism utilized internally is HTTP Redirection. CDN B returns the IP address of a Request Router so that the RR get the subsequent HTTP content request from the user agent.

3. CDN A proxy the response back to the user agent.

4. The user agent sends the content requests to the Request Router of CDN B. Based on local routing decision policy, e.g. whether content hits take the highest priority or location proximity takes the highest priority, the Request Router selects a delivery node to serve the user agent and returns an HTTP 302 message to redirect the content request.

5. The user agent performs a DNS lookup for the hostname of the delivery node and gets the IP address of the node.

6. The user agent requests the content from CDN B’s delivery node. The node contains the content, so it sends the content to the user agent.

3.1.1.2. DNS Redirection utilized by Downstream CDN

This example illustrates the CaseNo2 of Table 1. Based on local policy, the Upstream CDN and the Downstream CDN both utilize the DNS Redirection with the user agent. As the Downstream CDN cannot get the user agent’s location information through the DNS request forwarded by the Upstream CDN, in this case, the DNS resolver of the Downstream CDN is configured to return a CNAME of the RR to make it receive another DNS query request sent by the user agent/local DNS with information of the user’s location. Again, the distinguished domain name of the Downstream CDN is "cdni.op-b.example".

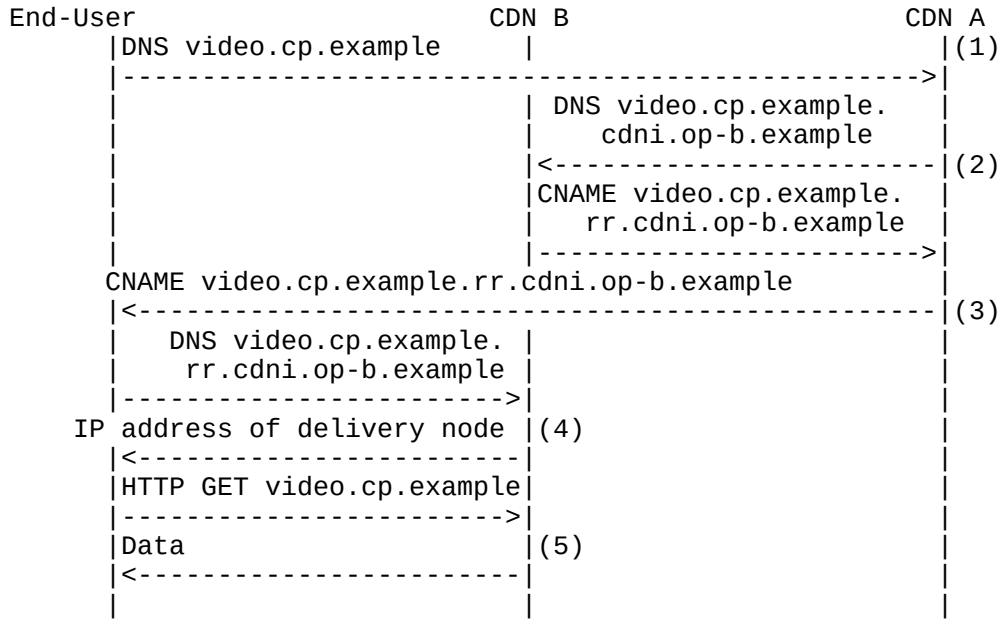


Figure 3 DNS based CDNI Recursive Request Routing 2

1. A Request Routing System of CDN A processes the DNS request for its customer based on the domain video.cp.example and recognizes that the end-user is best served by another CDN, specifically CDN B. Based on the pre-configured distinguished domain name of CDN B and rules that have been negotiated with CDN B that describe how to construct a domain name contained in a DNS query request over RRI, the Request Routing System changes the domain name to CDN B's domain name accompanying with the CP's domain information and acts as a proxy server forwarding the DNS request to CDN B.

   Note: For simplicity, the local DNS invoked in the procedure is not shown.

2. CDN B recognizes that the request is from a peer CDN rather than a user agent by the distinguished domain name. Based on the local policy, CDN B determines that the routing mechanism utilized internally is DNS Redirection. CDN B returns the CNAME of a Request Router so that the user agent will send another DNS query request to get the user agent's location information.

3. CDN A proxy the response back to the user agent.

4. The user agent sends the content requests to the Request Router of CDN B based on DNS with the CNAME of the RR. Based on local routing decision policy, the Request Router selects a delivery node to serve the user agent and returns IP address of the node.

5. The user agent requests the content from CDN B's delivery node, the node holds the content at the time and send the content to the user agent.

3.1.2. HTTP based Request Routing Protocol

This example illustrates the CaseNo4 of Table 1. Based on local policy, the Upstream CDN and the Downstream CDN both utilize the HTTP Redirection with the user agent. The Upstream CDN shall provide as much information as possible to the Downstream to assist making the routing decision. For example, it includes the content URI and the user's location information etc.

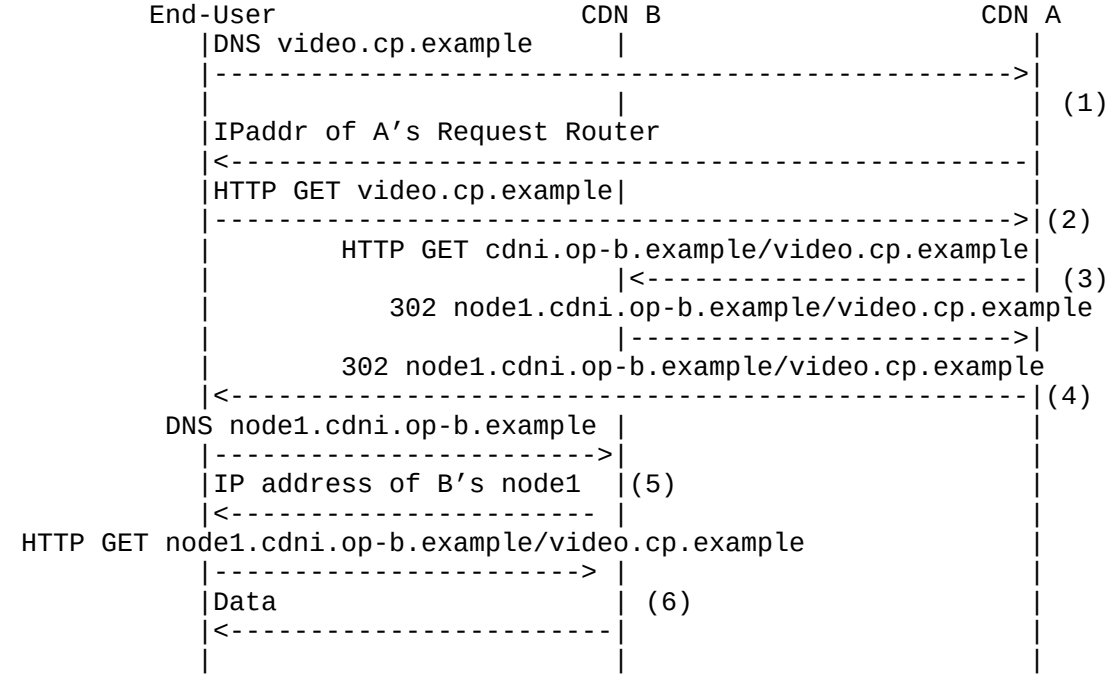


Figure 4 HTTP protocol based CDNI Recursive Request Routing

1. A DNS resolver for CDN A processes the DNS request for its customer based on the domain video.cp.example. Based on local policy, HTTP Redirection is adopted for this request. Hence it returns the IP address of a Request Router in CDN A.

2. A Request Router of CDN A processes the user agent's content request and recognizes that the end-user is best served by another CDN, specifically CDN B. Based on pre-configuration or other means of discovery, the Request Router pushes the distinguished domain name of CDN B onto the original URL and acts as a proxy server forwarding the request to CDN B's Request Router. It also appends an X-Forwarded-For header into the request with the value set to the user's IP address extracted from the IP package header of the HTTP request it received.

3. Based on local routing decision policy, e.g. whether content hits take the highest priority or location proximity takes the highest priority, the Request Router of CDN B select a delivery node for the end user and returns an HTTP 302 message to redirect the content request to the node.

4. CDN A proxies the response back to the user agent.

5. The user agent performs a DNS lookup for the hostname of the delivery node and gets the IP address of the node.

6. The user agent requests the content from CDN B's delivery node. Since the node holds the content, it sends the content to the user agent.

3.2. Capability Information Advertising

Besides forwarding a routing request from the Upstream CDN to the Downstream CDN at the request routing time, another important function of the Request Routing Protocol in Figure 1 is to advertis the capability information of the Downstream CDN to the Upstream CDN.

The Downstream CDN may advertise its delivery capability to the Upstream CDN de-coupled with the routing request itself, during a periodic interval, e.g. every 5 minutes. This is called "report mode" in this document. Another one is called "query mode", means the Upstream CDN does not hold sufficient information to decide which Downstream CDN is most appropriate to deliver the content for the end user. In query mode,it then acquires the capability information from the Downstream CDN dynamically before make its routing decision.

HTTP/1.1 [RFC2616] protocol is used for capability advertising. The detailed capability information description and message definition is described in section 5 of this document.

Note: Although Iterative Request Routing is not discussed in this document, however, the capability information advertising procedures specified are also applicable to Iterative Request Routing.

4. Protocol Specification

This section specifies how the Request Routing Protocol enables the Downstream CDN to advertise its capability to the Upstream CDN and to enable the Upstream CDN acting as a proxy server to forward the routing request from a user agent to the Downstream CDN.

4.1.1. Recursive Request Routing

4.1.1.1. DNS based Request Routing Protocol

4.1.1.1.1. Upstream CDN Behavior

Upon receiving a DNS query request from a user agent, the Request Routing System of the Upstream CDN SHALL first determine a routing mechanism according to local policy. In case of DNS Redirection is leveraged, based on the local routing policy, if it is aware that the user is best served by another CDN, the Upstream CDN SHALL select a Downstream CDN and forward the DNS request to the Downstream CDN. When HTTP Redirection is adopted, the Upstream CDN SHALL respond with the address of a Request Router of the Upstream CDN.

The QNAME contained in the DNS query request which is forwarded to the Downstream CDN SHALL be constructed by the rules negotiated by the two interconnected CDNs and based on the distinguished domain name of the Downstream CDN.

Upon receiving a response from the Downstream CDN, the Request Routing System of the Upstream CDN shall forward it back to the user agent with the DNS payload unchanged.

4.1.1.1.2. Downstream CDN Behavior

Upon receiving a DNS query request, the Downstream CDN SHALL first identify that this is a request for CDNI based on the distinguished domain name contained in the query request, and extracts the content provider's domain name. It then SHALL determine a routing mechanism according to local routing policy.

Note: The local routing policy may take into account the CP's policy if existed identified by the CP's domain name.

In case of DNS Redirection, it SHALL select a request router and return a response containing a CNAME with the hostname of the request router.

In case of HTTP Redirection, it SHALL select a request router and return a response containing the IP address of the Request Router.

4.1.1.2. HTTP based Request Routing Protocol

4.1.1.2.1. Upstream CDN Behavior

Upon receiving an HTTP GET request from a user agent for specific content, based on the local routing policy, if it is determined that the user is best served by another CDN, the Request Router of the Upstream CDN SHALL select a Downstream CDN for the end user, insert an X-Forwarded-For header into the request with the value set to the User Agent's IP address, augment the original URL with the distinguished domain name of the Downstream CDN in front of it, and then forward the request to the elected Downstream CDN.

After receiving an HTTP "302" redirection response from the Downstream CDN, the Upstream CDN SHALL forward it back to the user agent.

4.1.1.2.2. Downstream CDN Behavior

Upon receiving an HTTP GET request, the Downstream CDN SHALL select a delivery node for the user based on the local routing policy.
If the user's location information is required to make the routing decision, it SHALL obtain that from an X-Forwarded-For header if this hearder exists. The Downstream CDN SHALL then return a response with a 302 Redirection message. The Location header's value is constructed by truncating the CDN B's domain name from the original URL in the request it received, and inserting

the host name of the selected delivery node onto the front of the remaining URL.

4.1.2. Iterative Request Routing

Note: Whether any content relative to Iterative Request Routing should be added here is to be determined by the CDNI working group.

4.2. Capability Information Advertising

4.2.1. Capability information description

The Downstream CDN exposes capability information to an Upstream CDN on Request Routing Interface to facilitate CDN selection among other functions. The exposure should be of appropriate granularity to ensure the self-administrative nature of Downstream CDN.

The following information in Table 2 is considered for capability exchange.

Name	Type	Value	Description
IPVersion	ENUM,4 byte	1:IPV4;2:IPV6	IP address version
Service-Status	ENUM,4 byte	1:in-service;2:out-of-service	CDNI service status
MaxAcquisitionBW	UNIT32	Integer starts from zero. Unit:Mbps	Concurrent maximal availble bandwidth for content acquisition
UsedAcquisitionBW	UNIT32	Integer starts from zero. Unit:Mbps	Concurrent used bandwidth for content acquistion
MaxDeliveryBW	UNIT32	Integer starts from zero. Unit:Mbps	Concurrent maximal availble bandwidth for content delivery
UsedDeliveryBW	UNIT32	Integer starts from zero. Unit:Mbps	Concurrent used bandwidth for content delivery
Delivery-Protocol	List	A list of protocols,e.g.HTTP,RTSP	Supported delivery protocols
Coverage	List	Coverage represented by Contry, State and City combination	CDN coverage

Table 2 capability information description

4.2.2. Message description

The HTTP/1.1 protocol is used for capability advertising.

The HTTP request is HTTP POST for Report mode and HTTP GET for Query mode respectively.

The request URI in both modes conforms to [RFC3986]. The URI format in this document is as follows: HTTP://<host>/<url-path>, where the <host> specifies a destination, and the <url-path> conveys the message name.

The message body representation specified in this document is JavaScript Object Notation(JSON).

4.2.2.1. Report mode

The Downstream CDN issues an HTTP POST message to the Upstream CDN to report its capability information.

The message name in the request URI is "CdniCapReport".

The Content-Type header field is "application/json".

The message body includes capability information.

Upon successful receipt of the POST request, the Upstream CDN responds with a 200 OK message.

4.2.2.2. Query mode

The Upstream CDN issues a HTTP GET message to a Downstream CDN to query its capability information.

The message name in the request URI is "CdniCapQuery".

Upon successful receipt of the GET request, the Downstream CDN responds a 200 OK message with its capability information.

The Content-Type header field for the response is "application/json".

4.2.3. Message examples

4.2.3.1. Report mode

The POST request and corresponding response are illustrated below.

Request example (Downstream CDN to Upstream CDN):

```
POST http://contact-address.ucdn.example/CdniCapReport HTTP/1.1
Content-Type: application/json
Content-Length: 350

{
     "IPVersion":1,
     "ServiceStatus":1,
     "MaxAquisitionBW":10000,
     "UsedAquisitionBW":2000,
     "MaxDeliveryBW":20000,
     "UsedDeliveryBW":5000,
     "DeliveryProtocol":["HTTP","RSTP"],

     "Coverage":
     [
         {
              "Country":"China",
               "State":[
                    {
                        "Name":"Beijing",
                        "City":["CityA","CityB"]
                    },
                    {
                        "Name":"Shanghai",
                        "City":["CityX"]
                    }
                ]
         },
```

```
            {
                "Country":"US",
                 "State":[
                    {
                         "Name":"California",
                         "City":["CityY"]
                     }
                ]
            }
       ]
    }
```

Response example:

HTTP/1.1 200 OK

4.2.3.2. Query mode

The GET request and corresponding response are illustrated below.

Request example (Upstream CDN to Downstream CDN):

GET http://contact-address.dcdn.example/CdniCapQuery HTTP/1.1

Response example:

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 350

The content of message body is the same as that of POST message illustrated in section 5.2.4.1.

5. Security Considerations

In HTTP based Recursive Request Routing, the end user’s web browsers will not send cookies if the content request is redirected to a URL in a different domain rather than the original CP’s domain, e.g. the Downstream CDN’s domain. If the browser is expected to send

any cookies associated with the original CP's domain, this will cause problem that the CP's policy is not enforced by the CDN.

The section 5.2 of draft [I-D.draft-peterson-cdni-strawman] has discussed a similar question and given a solution.

6. IANA Considerations

If the approach described in this document is adopted, we would request that IANA allocate the message name "CdniCapReport" and "CdniCapQuery" in the HTTP Parameters registry.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3986] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax and Semantics", RFC 3986, January 2005.

7.2. Informative References

[I-D.draft-cdni-use-cases]
"Use Cases for Content Delivery Network Interconnection", Gilles Bertrand, Stephan Emile, Grant Watson, Trevor Burbridge, Philip Eardley, Kevin Ma, 22-Sep-11, <draft-ietf-cdni-use-cases-00.txt>.

[I-D.draft-cdni-problem-statement]
"Content Distribution Network Interconnection (CDNI) Problem Statement", Ben Niven-Jenkins, Francois Faucheur, Nabil Bitar, 9-Sep-11, <draft-ietf-cdni-problem-statement-00.txt>.

[I-D.draft-cdni-requirements]
"Content Distribution Network Interconnection (CDNI) Requirements", Kent Leung, Yiu Lee, 9-Sep-11, <draft-ietf-cdni-requirements-00.txt>.

[I-D.draft-peterson-cdni-strawman]
"A Simple Approach to CDN Interconnection", Larry Peterson, John Hartman, 18-May-11,<draft-peterson-cdni-strawman-01.txt,.pdf>.

[I-D.davie-cdni-framework]
Davie, B. and L. Peterson, "Framework for CDN Interconnection", draft-davie-cdni-framework-00 (work in progress), July 2011.

8. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Xiaoyan He
Huawei
B2, Huawei Industrial Base, 518129
China

Email: hexiaoyan@huawei.com

Jincheng Li
Huawei
B2, Huawei Industrial Base, 518129
China

Email: lijincheng@huawei.com

Spencer Dawkins
Huawei

Email: spencer.dawkins@huawei.com

Ge Chen
China Telecom
109 West Zhongshan Ave,Tianhe District,Guangzhou,P.R.C

Email: cheng@gsta.com