CDNI
Internet-Draft
Intended status: Standards Track
Expires: February 15, 2020

J. Seedorf
HFT Stuttgart - Univ. of Applied Sciences
Y. Yang
Tongji/Yale
K. Ma
Ericsson
J. Peterson
Neustar
X. Lin
J. Zhang
Tongji

August 14, 2019

Content Delivery Network Interconnection (CDNI) Request Routing: CDNI Footprint and Capabilities Advertisement using ALTO draft-ietf-alto-cdni-request-routing-alto-07

Abstract

The Content Delivery Networks Interconnection (CDNI) framework [RFC6707] defines a set of protocols to interconnect CDNs, to achieve multiple goals such as extending the reach of a given CDN to areas that are not covered by that particular CDN. One component that is needed to achieve the goal of CDNI described in [RFC7336] is the CDNI Request Routing Footprint & Capabilities Advertisement interface (FCI). [RFC8008] defines precisely the semantics of FCI and provides guidelines on the FCI protocol, but the exact protocol is explicitly outside the scope of that document. In this document, we follow the guidelines to define an FCI protocol using the Application-Layer Traffic Optimization (ALTO) protocol.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 15, 2020.

Table of Contents

1. Introduction

The ability to interconnect multiple content delivery networks (CDNs) has many benefits, including increased coverage, capability, and reliability. The Content Delivery Networks Interconnection (CDNI) framework [RFC6707] defines four interfaces to achieve interconnection of CDNs: (1) the CDNI Request Routing Interface; (2) the CDNI Metadata Interface; (3) the CDNI Logging Interface; and (4) the CDNI Control Interface.

Among the four interfaces, the CDNI Request Routing Interface provides key functions, as specified in [RFC6707]: "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." On a high level, the scope of the CDNI Request Routing Interface, therefore, contains two main tasks: (1) determining if the downstream CDN (dCDN) is willing to accept a delegated content request; (2) redirecting the content request coming from an upstream CDN (uCDN) to the proper entry point or entity in the downstream CDN.

Correspondingly, the request routing interface is broadly divided into two functionalities: (1) CDNI Footprint & Capabilities Advertisement interface (FCI); (2) CDNI Request Routing Redirection interface (RI). Since this document focuses on the first functionality, CDNI FCI, we will describe it in a more detailed way. CDNI FCI is an advertisement from a dCDN to a uCDN (push) or a query from a uCDN to a dCDN (pull) so that the uCDN knows whether it can redirect a particular user request to that dCDN.

A key component in defining CDNI FCI is defining objects describing the footprints and capabilities of a dCDN. Such objects are already in [RFC8008]. A protocol to transport and update such objects between a uCDN and a dCDN, however, is not defined. Hence, the scope of this document is to define such a protocol by introducing a new Application-Layer Traffic Optimization (ALTO) [RFC7285] service called "CDNI FCI Map Service".

There are multiple benefits in using ALTO as a transport protocol, as we discuss in Section 2.2.

The rest of this document is organized as follows. Section 2 provides non-normative background on both CDNI FCI and ALTO. Section 3 introduces the most basic service, called CDNI FCI Map, to realize CDNI FCI using ALTO. Section 4 demonstrates a key benefit of using ALTO: the ability to integrate CDNI FCI with ALTO network maps. Such integration provides a new granularity to describe footprints. Section 5 builds on filtered ALTO maps to introduce filtered CDNI FCI maps using capabilities so that a uCDN can get footprints with given capabilities instead of getting the full map which can be huge. Section 6 further shows a benefit of using ALTO: the ability to query footprint properties using ALTO unified properties. In this way, a uCDN can effectively fetch capabilities of some footprints in which it is interested. IANA and security considerations are discussed in Section 7 and Section 8 respectively.

Throughout this document, we use the terminology for CDNI defined in [RFC6707], [RFC8006], [RFC8008] and we use the terminology for ALTO defined in [RFC7285], [I-D.ietf-alto-unified-props-new].

2. Background

The design of CDNI FCI transport using ALTO depends on the understanding of both FCI semantics and ALTO. Hence, we start with a review of both.

2.1. Semantics of FCI Advertisement

The CDNI document on "Footprint and Capabilities Semantics" [RFC8008] defines the semantics of CDNI FCI, and provides guidance on what Footprint and Capabilities mean in a CDNI context and how a protocol solution should in principle look like. The definitions in [RFC8008] depend on [RFC8006]. Here we briefly summarize key related points of [RFC8008] and [RFC8006]. For a detailed discussion, the reader is referred to the RFCs.

- Footprint and capabilities are tied together and cannot be interpreted independently from each other. Hence, capabilities must be expressed on a per footprint basis. [RFC8008] integrates footprint and capabilities with an approach of "capabilities with footprint restrictions".

- Given that a large part of Footprint and Capabilities Advertisement will actually happen in contractual agreements, the semantics of CDNI Footprint and Capabilities advertisement refers to answering the following question: what exactly still needs to be advertised by the CDNI FCI? For instance, updates about temporal failures of part of a footprint can be useful information to convey via the CDNI request routing interface. Such information would provide updates on information previously agreed in contracts between the participating CDNs. In other words, the CDNI FCI is a means for a dCDN to provide changes/updates regarding a footprint and/or capabilities that it has prior agreed to serve in a contract with a uCDN. Hence, server push and incremental encoding will be necessary techniques.

- Multiple types of footprints (ipv4cidr, ipv6cidr, asn and countrycode) are defined in [RFC8006].

- A "Set of IP-prefixes" can contain both full IP addresses (i.e., a /32 for IPv4 and a /128 for IPv6) and IP prefixes with an arbitrary prefix length. There must also be support for multiple IP address versions, i.e., IPv4 and IPv6, in such a footprint.

- For all of these mandatory-to-implement footprint types, footprints can be viewed as constraints for delegating requests to a dCDN: A dCDN footprint advertisement tells the uCDN the limitations for delegating a request to the dCDN. For IP prefixes or ASN(s), the footprint signals to the uCDN that it should consider the dCDN a candidate only if the IP address of the request routing source falls within the prefix set (or ASN, respectively). The CDNI specifications do not define how a given uCDN determines what address ranges are in a particular ASN. Similarly, for country codes, a uCDN should only consider the dCDN

a candidate if it covers the country of the request routing source. The CDNI specifications do not define how a given uCDN determines the country of the request routing source. Multiple footprint constraints are additive, i.e., the advertisement of different types of footprint narrows the dCDN candidacy cumulatively.

- The following capabilities are defined as "base" capabilities; that is, they are required in all cases and therefore constitute mandatory capabilities to be supported by the CDNI FCI: (1) Delivery Protocol; (2) Acquisition Protocol; (3) Redirection Mode; (4) Capabilities related to CDNI Logging; (5) Capabilities related to CDNI Metadata.

2.2. ALTO Background and Benefits

Application-Layer Traffic Optimization (ALTO) [RFC7285] 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 that these applications cannot or have difficulty to measure themselves [RFC5693].

Originally, ALTO was motivated by optimizing cross-ISP traffic generated by P2P applications [RFC5693]. Recently, however, ALTO is also being considered for improving the request routing in CDNs [I-D.jenkins-alto-cdn-use-cases]. The CDNI problem statement explicitly mentions ALTO as a candidate protocol for "actual algorithms for selection of CDN or Surrogate by Request-Routing systems" [RFC6707].

The following reasons make ALTO a suitable candidate protocol for downstream CDN selection as part of CDNI request routing and in particular for an FCI protocol:

- ALTO is a protocol specifically designed to improve application layer traffic (and application layer connections among hosts on the Internet) by providing additional information to applications that these applications could not easily retrieve themselves. For CDNI, this is exactly the case: a uCDN wants to improve application layer CDN request routing by using dedicated information (provided by a dCDN) that the uCDN could not easily obtain otherwise. ALTO can help a uCDN to select a proper dCDN by first providing dCDNs' capabilities as well as footprints (see

Section 3) and then providing costs of surrogates in a dCDN by ALTO cost maps.

- The semantics of an ALTO network map is an exact match for the needed information to convey a footprint by a downstream CDN, in particular if such a footprint is being expressed by IP-prefix ranges. Please see Section 4.

- Security: Identifications between uCDNs and dCDNs are extremely important. ALTO maps can be signed and hence provide inherent integrity protection. Please see Section 8.

- RESTful-Design: The ALTO protocol has undergone extensive revisions in order to provide a RESTful design regarding the client-server interaction specified by the protocol. A CDNI FCI interface based on ALTO would inherit this RESTful design. Please see Section 3.

- Error-handling: The ALTO protocol has undergone extensive revisions in order to provide sophisticated error-handling, in particular regarding unexpected cases. A CDNI FCI interface based on ALTO would inherit this thought-through and mature error-handling. Please see Section 5.

- Filtered map service: The ALTO map filtering service would allow a uCDN to query only for parts of an ALTO map. For example, filtered unified property map service can enable a uCDN to query properties of a part of footprints in an effective way (see Section 6).

- Server-initiated Notifications and Incremental Updates: When the footprint or the capabilities of a downstream CDN change (i.e., unexpectedly from the perspective of an upstream CDN), server-initiated notifications would enable a dCDN to directly inform an upstream CDN about such changes. Consider the case where - due to failure - part of the footprint of the dCDN is not functioning, i.e., the CDN cannot serve content to such clients with reasonable QoS. Without server-initiated notifications, the uCDN might still use a very recent network and cost map from dCDN, and therefore redirect requests to dCDN which it cannot serve. Similarly, the possibility for incremental updates would enable efficient conveyance of the aforementioned (or similar) status changes by the dCDN to the uCDN. The newest design of ALTO supports server pushed incremental updates [I-D.ietf-alto-incr-update-sse].

- Content Availability on Hosts: A dCDN might want to express CDN capabilities in terms of certain content types (e.g., codecs/formats, or content from certain content providers). The new

endpoint property for ALTO would enable a dCDN to make such information available to an upstream CDN. This would enable a uCDN to determine if a given dCDN actually has the capabilities for a given request with respect to the type of content requested.

- Resource Availability on Hosts or Links: The capabilities on links (e.g. maximum bandwidth) or caches (e.g. average load) might be useful information for an upstream CDN for optimized downstream CDN selection. For instance, if a uCDN receives a streaming request for content with a certain bitrate, it needs to know if it is likely that a dCDN can fulfill such stringent application-level requirements (i.e., can be expected to have enough consistent bandwidth) before it redirects the request. In general, if ALTO could convey such information via new endpoint properties, it would enable more sophisticated means for downstream CDN selection with ALTO. ALTO Path Vector Extension [I-D.ietf-alto-path-vector] is designed to allow ALTO clients to query information such as capacity regions for a given set of flows.

3. CDNI FCI Service

The ALTO protocol is based on an ALTO Information Service Framework which consists of several services, where all ALTO services are "provided through a common transport protocol, messaging structure and encoding, and transaction model" [RFC7285]. The ALTO protocol specification [RFC7285] defines several such services, e.g., the ALTO map service.

This document defines a new ALTO Service called "CDNI FCI Service" which conveys JSON objects of media type "application/alto-cdnifci+json". These JSON objects are used to transport BaseAdvertisementObject objects defined in [RFC8008]; this document specifies how to transport such BaseAdvertisementObject objects via the ALTO protocol with the ALTO "CDNI FCI Service". Similar to other ALTO services, this document defines the ALTO information resource for the "CDNI FCI Service" as follows.

3.1. Media Type

The media type of the CDNI FCI resource is "application/alto-cdnifci+json".

3.2. HTTP Method

A CDNI FCI resource is requested using the HTTP GET method.

3.3. Accept Input Parameters

None.

3.4. Capabilities

None.

3.5. Uses

The "uses" field SHOULD NOT appear unless the CDNI FCI resource depends on some ALTO information resources. If the CDNI FCI resource has some dependent resources, the resource IDs of its dependent resources MUST be included into the "uses" field. This document only defines one potential dependent resource for the CDNI FCI resource. See Section 4 for details of when and how to use it. The future documents may extend the CDNI FCI resource and allow other dependent resources.

3.6. Response

The "meta" field of a CDNI FCI response MUST include the "vtag" field defined in Section 10.3 of [RFC7285]. This field provides the version of the retrieved CDNI FCI map.

If a CDNI FCI response depends on an ALTO information resource, it MUST include the "dependent-vtags" field, whose value is an array to indicate the version tags of the resources used, where each resource is specified in "uses" of its IRD entry.

The data component of an ALTO CDNI FCI response is named "cdni-fci", which is a JSON object of type CDNIFCIData:

```
object {
    CDNIFCIData cdni-fci;
} InfoResourceCDNIFCI : ResponseEntityBase;

object {
    BaseAdvertisementObject capabilities<1..*>;
} CDNIFCIData;
```

Specifically, a CDNIFCIData object is a JSON object that includes only one property named "capabilities", whose value is an array of BaseAdvertisementObject objects.

The syntax and semantics of BaseAdvertisementObject are well defined in Section 5.1 of [RFC8008]. A BaseAdvertisementObject object includes multiple properties, including capability-type, capability-

value and footprints, where footprints are defined in Section 4.2.2.2 of [RFC8006].

To be self-contained, we give a non-normative specification of BaseAdvertisementObject below. As mentioned above, the normative specification of BaseAdvertisementObject is in [RFC8008]

```
object {
    JSONString capability-type;
    JSONValue capability-value;
    Footprint footprints<0..*>;
} BaseAdvertisementObject;

object {
    JSONString footprint-type;
    JSONString footprint-value<1..*>;
} Footprint;
```

For each BaseAdvertisementObject, the ALTO client MUST interpret footprints appearing multiple times as if they appeared only once. If footprints in a BaseAdvertisementObject is null or empty or not appearing, the ALTO client MUST understand that the capabilities in this BaseAdvertisementObject have the "global" coverage.

Note: Further optimization of BaseAdvertisement objects to effectively provide the advertisement of capabilities with footprint restrictions is certainly possible. For example, these two examples below both describe that the dCDN can provide capabilities ["http/1.1", "https/1.1"] for the same footprints. However, the latter one is smaller in its size.

EXAMPLE 1

```
{
  "meta" : {...},
  "cdni-fci": {
    "capabilities": [
      {
        "capability-type": "FCI.DeliveryProtocol",
        "capability-value": {
          "delivery-protocols": [
            "http/1.1"
          ]
        },
        "footprints": [
          <Footprint objects>
        ]
      },
      {
```

```
              "capability-type": "FCI.DeliveryProtocol",
              "capability-value": {
                "delivery-protocols": [
                  "https/1.1"
                ]
              },
              "footprints": [
                <Footprint objects>
              ]
            }
          ]
        }
      }
```

EXAMPLE 2

```
      {
        "meta" : {...},
        "cdni-fci": {
          "capabilities": [
            {
              "capability-type": "FCI.DeliveryProtocol",
              "capability-value": {
                "delivery-protocols": [
                  "https/1.1",
                  "http/1.1"
                ]
              },
              "footprints": [
                <Footprint objects>
              ]
            }
          ]
        }
      }
```

Since such optimizations are not required for the basic interconnection of CDNs, the specifics of such mechanisms are outside the scope of this document.

3.7. Examples

3.7.1. IRD Example

Below is the information resource directory (IRD) of a simple, example ALTO server. The server provides both base ALTO information resources (e.g., network maps) and CDNI FCI related information resources (e.g., CDNI FCI resource), demonstrating a single, integrated environment.

Specifically, the IRD announces two network maps, one CDNI FCI resource without dependency, one CDNI FCI resource depending on a network map, one filtered CDNI FCI resource to be defined in Section 5, one unified property map including "cdni-fci-capabilities" as its entity property, one filtered unified property map including "cdni-fci-capabilities" and "pid" as its entity properties, and two update stream services (one for updating CDNI FCI resources, and the other for updating property maps).

```
 GET /directory HTTP/1.1
 Host: alto.example.com
 Accept: application/alto-directory+json,application/alto-error+json

 {
   "meta" : { ... },
   "resources": {
     "my-default-network-map": {
       "uri" : "http://alto.example.com/networkmap",
       "media-type" : "application/alto-networkmap+json"
     },
     "my-eu-netmap" : {
       "uri" : "http://alto.example.com/myeunetmap",
       "media-type" : "application/alto-networkmap+json"
     },
     "my-default-cdnifci": {
       "uri" : "http://alto.example.com/cdnifci",
       "media-type": "application/alto-cdnifci+json"
     },
     "my-filtered-cdnifci" : {
       "uri" : "http://alto.example.com/cdnifci/filtered",
       "media-type" : "application/alto-cdnifci+json",
       "accepts" : "application/alto-cdnifcifilter+json",
       "uses" : [ "my-default-cdnifci" ]
     },
     "my-cdnifci-with-pid-footprints": {
       "uri" : "http://alto.example.com/networkcdnifci",
       "media-type" : "application/alto-cdnifci+json",
       "uses" : [ "my-eu-netmap" ]
     },
     "cdnifci-property-map" : {
       "uri" : "http://alto.example.com/propmap/full/cdnifci",
       "media-type" : "application/alto-propmap+json",
       "uses": [ "my-default-cdni" ],
       "capabilities" : {
         "mappings": {
           "ipv4": [ "my-default-cdni.cdni-fci-capabilities" ],
           "ipv6": [ "my-default-cdni.cdni-fci-capabilities" ],
           "countrycode": [
```

```
            "my-default-cdni.cdni-fci-capabilities" ],
          "asn": [ "my-default-cdni.cdni-fci-capabilities" ],
        }
      }
    },
    "filtered-cdnifci-property-map" : {
      "uri" : "http://alto.example.com/propmap/lookup/cdnifci-pid",
      "media-type" : "application/alto-propmap+json",
      "accepts" : "application/alto-propmapparams+json",
      "uses": [ "my-default-cdni", "my-default-network-map" ],
      "capabilities" : {
        "mappings": {
          "ipv4": [ "my-default-cdni.cdni-fci-capabilities",
                    "my-default-network-map.pid" ],
          "ipv6": [ "my-default-cdni.cdni-fci-capabilities",
                    "my-default-network-map.pid" ],
          "countrycode": [
            "my-default-cdni.cdni-fci-capabilities" ],
          "asn": [ "my-default-cdni.cdni-fci-capabilities" ],
        }
      }
    },
    "update-my-cdni-fci" : {
      "uri": "http:///alto.example.com/updates/cdnifci",
      "media-type" : "text/event-stream",
      "accepts" : "application/alto-updatestreamparams+json",
      "uses" : [
        "my-default-network-map",
        "my-eu-netmap",
        "my-default-cdnifci",
        "my-filtered-cdnifci"
        "my-cdnifci-with-pid-footprints"
      ],
      "capabilities" : {
        "incremental-change-media-types" : {
         "my-default-network-map" : "application/json-patch+json",
         "my-eu-netmap" : "application/json-patch+json",
         "my-default-cdnifci" :
         "application/merge-patch+json,application/json-patch+json",
         "my-filtered-cdnifci" :
         "application/merge-patch+jso,application/json-patch+json",
         "my-cdnifci-with-pid-footprints" :
         "application/merge-patch+json,application/json-patch+json"
        }
      }
    },
    "update-my-props": {
      "uri" : "http://alto.example.com/updates/properties",
```

```
          "media-type" : "text/event-stream",
          "uses" : [
            "cdnifci-property-map",
            "filtered-cdnifci-property-map"
          ],
          "capabilities" : {
            "incremental-change-media-types": {
             "cdnifci-property-map" :
             "application/merge-patch+json,application/json-patch+json",
             "filtered-cdnifci-property-map":
             "application/merge-patch+json,application/json-patch+json"
            }
          }
        }
      }
    }
```

3.7.2. Basic Example

In this example, we demonstrate a simple CDNI FCI resource; this resource does not depend on other resources. There are three BaseAdvertisementObjects in this map and these objects' capabilities are http/1.1 delivery protocol, [http/1.1, https/1.1] delivery protocol and https/1.1 acquisition protocol respectively.

```
  GET /cdnifci HTTP/1.1
  Host: alto.example.com
  Accept: application/alto-cdnifci+json,
          application/alto-error+json

  HTTP/1.1 200 OK
  Content-Length: XXX
  Content-Type: application/alto-cdnifci+json
  {
    "meta" : {
      "vtag": {
        "resource-id": "my-default-cdnifci",
        "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
      }
    },
    "cdni-fci": {
      "capabilities": [
        {
          "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": [
              "http/1.1"
            ]
```

```
            },
            "footprints": [
              <Footprint objects>
            ]
          },
          {
            "capability-type": "FCI.DeliveryProtocol",
            "capability-value": {
              "delivery-protocols": [
                "https/1.1",
                "http/1.1"
              ]
            },
            "footprints": [
              <Footprint objects>
            ]
          },
          {
            "capability-type": "FCI.AcquisitionProtocol",
            "capability-value": {
              "acquisition-protocols": [
                "https/1.1"
              ]
            },
            "footprints": [
              <Footprint objects>
            ]
          }
        ]
      }
    }
```

3.7.3. Incremental Updates Example

A benefit of using ALTO to provide CDNI FCI maps is that such maps can be updated using ALTO incremental updates. Below is an example that also shows the benefit of having both JSON merge patch and JSON patch to encode updates.

At first, an ALTO client requests updates for "my-default-cdnifci", and the ALTO server returns the "control-uri" followed by the full CDNI FCI response. Then when there is a change in the delivery-protocols in that ‘http/2‘ is removed (from http/1.1 and http/2 to only http/1.1) due to maintenance of the http/2 clusters, the ALTO server uses JSON merge patch to encode the change and pushes the change to the ALTO client. Later on, the ALTO server notifies the ALTO client that "ipv4:192.0.2.0/24" is added into the footprint for

delivery-protocol http/1.1 by sending the change encoded by JSON patch to the ALTO client.

```
POST /updates/cdnifci HTTP/1.1
Host: alto.example.com
Accept: text/event-stream,application/alto-error+json
Content-Type: application/alto-updatestreamparams+json
Content-Length: ###

{ "add": {
    "my-cdnifci-stream": {
        "resource-id": "my-default-cdnifci"
    }
}

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream

event: application/alto-updatestreamcontrol+json
data: {"control-uri":
data: "http://alto.example.com/updates/streams/3141592653589"}

event: application/alto-cdnifci+json,my-default-cdnifci
data: { ... full CDNI FCI map ... }

event: application/merge-patch+json,my-default-cdnifci
data: {
data:   "meta": {
data:     "vtag": {
data:       "tag": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
data:     }
data:   },
data:   "cdni-fci": {
data:     "capabilities": [
data:       {
data:         "capability-type": "FCI.DeliveryProtocol",
data:         "capability-value": {
data:           "delivery-protocols": [
data:             "http/1.1"
data:           ]
data:         },
data:         "footprints": [
data:           <Footprint objects in only http/1.1>
data:         ]
data:       }
data:     ]
data:   }
```

```
     data: }

     event: application/json-patch+json,my-default-cdnifci
     data: [
     data:   {
     data:     "op": "replace",
     data:     "path": "/meta/vtag/tag",
     data:     "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
     data:   },
     data:   { "op": "add",
     data:     "path": "/cdni-fci/capabilities/0/footprints/-",
     data:     "value": "ipv4:192.0.2.0/24"
     data:   }
     data: ]
```

4. CDNI FCI Service using ALTO Network Map

4.1. Network Map Footprint Type: altopid

The ALTO protocol defines a concept called PID to represent a group of IPv4 or IPv6 addresses which can be applied the same management policy. The PID is an alternative to the pre-defined CDNI footprint types (i.e., ipv4cidr, ipv6cidr, asn, and countrycode).

Specifically, a CDNI FCI resource can depend on an ALTO network map resource and use a new CDNI Footprint Type called "altopid" to compress its CDNI Footprint Payload.

"altopid" footprint type indicates that the corresponding footprint value is a list of PIDNames as defined in [RFC7285]. These PIDNames are references of PIDs in a network map resource. Hence a CDNI FCI with "altopid" footprints depends on a network map. For such a CDNI FCI map, the resource id of its dependent network map MUST be included in the "uses" field of its IRD entry, and the "dependent-vtag" field with a reference to this network map MUST be included in its response (see the example in Section 4.2.3).

4.2. Examples

4.2.1. IRD Example

We use the same IRD example given in Section 3.7.1.

4.2.2. ALTO Network Map for CDNI FCI Footprints Example

Below is an example network map whose resource id is "my-eu-netmap", and this map is referenced by the CDNI FCI example in Section 4.2.3.

```
GET /networkmap HTTP/1.1
Host: http://alto.example.com/myeunetmap
Accept: application/alto-networkmap+json,application/alto-error+json

HTTP/1.1 200 OK
Content-Length: XXX
Content-Type: application/alto-networkmap+json

{
  "meta" : {
    "vtag": [
      {"resource-id": "my-eu-netmap",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
      }
    ]
  },
  "network-map" : {
    "south-france" : {
      "ipv4" : [ "192.0.2.0/24", "198.51.100.0/25" ]
    },
    "germany" : {
      "ipv4" : [ "192.0.3.0/24"]
    }
  }
}
```

4.2.3. ALTO PID Footprints in CDNI FCI

In this example, we show a CDNI FCI resource that depends on a network map described in Section 4.2.2.

```
GET /networkcdnifci HTTP/1.1
Host: alto.example.com
Accept: application/alto-cdnifci+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: 618
Content-Type: application/alto-cdnifci+json

{
  "meta" : {
    "dependent-vtags" : [
      {
        "resource-id": "my-eu-netmap",
        "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
      }
    ]
  },
  "cdni-fci": {
    "capabilities": [
      { "capability-type": "FCI.DeliveryProtocol",
        "capability-value": [
          "http/1.1"
        ]
      },
      { "capability-type": "FCI.DeliveryProtocol",
        "capability-value": [
          "https/1.1"
        ],
        "footprints": [
          { "footprint-type": "altopid",
            "footprint-value": [
              "germany",
              "south-france"
            ]
          }
        ]
      }
    ]
  }
}
```

4.2.4. Incremental Updates Example

In this example, the ALTO client is interested in changes of "my-cdnifci-with-pid-footprints". Considering two changes, the first one is to change footprints of http/1.1 Delivery Protocol capability, and the second one is to remove "south-france" from the footprints of https/1.1 delivery protocol capability.

```
POST /updates/cdnifci HTTP/1.1
Host: alto.example.com
Accept: text/event-stream,application/alto-error+json
```

```
Content-Type: application/alto-updatestreamparams+json
Content-Length: ###

{ "add": {
    "my-network-map-cdnifci-stream": {
        "resource-id": "my-cdnifci-with-pid-footprints"
    }
}

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream

event: application/alto-updatestreamcontrol+json
data: {"control-uri":
data: "http://alto.example.com/updates/streams/3141592653590"}

event: application/alto-cdnifci+json,my-fci-stream
data: { ... full CDNI FCI resource ... }

event: application/merge-patch+json,my-fci-stream
data: {
data:   "meta": {
data:     "dependent-vtags" : [
data:      {
data:        "resource-id": "my-eu-netmap",
data:        "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
data:      }
data:     ],
data:     "vtag": {
data:       "tag": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
data:     }
data:   },
data:   {
data:     "capability-type": "FCI.DeliveryProtocol",
data:     "capability-value": {
data:       "delivery-protocols": [
data:         "http/1.1"
data:       ]
data:     },
data:     "footprints": [
data:       <All footprint objects in http/1.1>
data:     ]
data:   }
data: }

event: application/json-patch+json,my-fci-stream
data: [
```

```
data:   {
data:     "op": "replace",
data:     "path": "/meta/vtag/tag",
data:     "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
data:   },
data:   { "op": "remove",
data:     "path": "/cdni-fci/capabilities/2/footprints/0/
data:       footprint-value/1",
data:   }
data: ]
```

5. Filtered CDNI FCI using Capabilities

Section 3 and Section 4 describe CDNI FCI Service which can be used to enable a uCDN to get capabilities with footprints constrains from dCDNs. However, always getting full CDNI FCI resources from dCDNs is very inefficient, hence we introduce a new service named "Filtered CDNI FCI Service" to allow a client to filter a CDNI FCI resource using a client-given set of capabilities. For each entry of the CDNI FCI response, only if the entry contains at least one of the client-given capabilities will it be returned to the client. The relationship between a filtered CDNI FCI resource and a CDNI FCI resource is similar to the relationship between a filtered network/cost map and a network/cost map.

5.1. Media Type

A filtered CDNI FCI resource uses the same media type defined for the CDNI FCI resource in Section 3.1.

5.2. HTTP Method

A filtered CDNI FCI resource is requested using the HTTP POST method.

5.3. Accept Input Parameters

The input parameters for a filtered CDNI FCI resource are supplied in the entity body of the POST request. This document specifies the input parameters with a data format indicated by the media type "application/alto-cdnifcifilter+json" which is a JSON object of type ReqFilteredCDNIFCI, where:

```
object {
    JSONString capability-type;
    JSONValue capability-value;
} CDNIFCICapability;

object {
    [CDNIFCICapability cdni-fci-capabilities<0..*>;]
} ReqFilteredCDNIFCI;
```

with fields:

capability-type: The same as Base Advertisement Object's capability-type defined in Section 5.1 of [RFC8008].

capability-value: The same as Base Advertisement Object's capability-value defined in Section 5.1 of [RFC8008].

cdni-fci-capabilities: A list of CDNI FCI capabilities defined in Section 5.1 of [RFC8008] for which footprints are to be returned. If a list is empty or not appearing, the ALTO server MUST interpret it as a request for the full CDNI FCI resource. The ALTO server MUST interpret entries appearing in a list multiple times as if they appeared only once. If the ALTO server does not define any footprints for a CDNI capability, it MUST omit this capability from the response.

5.4. Capabilities

None.

5.5. Uses

The resource ID of the CDNI FCI resource based on which the filtering is performed.

5.6. Response

The response MUST indicate an error, using ALTO protocol error handling specified in Section 8.5 of the ALTO protocol [RFC7285], if the request is invalid.

Specifically, a filtered CDNI FCI request is invalid if:

- the value of "capability-type" is null;

- the value of "capability-value" is null;

- the value of "capability-value" is inconsistent with "capability-type".

When a request is invalid, the ALTO server MUST return an "E_INVALID_FIELD_VALUE" error defined in Section 8.5.2 of [RFC7285], and the "value" field of the error message SHOULD indicate this CDNI FCI capability.

The ALTO server returns a filtered CDNI FCI resource for a valid request. The format of a filtered CDNI FCI resource is the same as an full CDNI FCI resource (See Section 3.6.)

The returned CDNI FCI resource MUST contain only BaseAdvertisementObject objects whose CDNI capability object is the superset of one of CDNI capability object in "cdni-fci-capabilities". Specifically, that a CDNI capability object A is the superset of another CDNI capability object B means that these two CDNI capability objects have the same capability type and mandatory properties in capability value of A MUST include mandatory properties in capability value of B semantically. See Section 5.7.2 for a concrete example.

The version tag included in the "vtag" field of the response MUST correspond to the full CDNI FCI resource from which the filtered CDNI FCI resource is provided. This ensures that a single, canonical version tag is used independently of any filtering that is requested by an ALTO client.

5.7. Examples

5.7.1. IRD Example

We use the same IRD example by Section 3.7.1.

5.7.2. Basic Example

This example filters the full CDNI FCI resource in Section 3.7.2 by selecting only http/1.1 delivery protocol capability. Only the first two BaseAdvertisementObjects in the full resource will be returned because the first object's capability is http/1.1 delivery protocol and the second object's capability is http/1.1 and https/1.1 delivery protocols which is the superset of http/1.1 delivery protocol.

```
POST /cdnifci/filtered HTTP/1.1
HOST: alto.example.com
Content-Type: application/cdnifilter+json
Accept: application/alto-cdnifci+json

{
```

```
      "cdni-fci-capabilities": [
        {
          "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": [
              "http/1.1"
            ]
          }
        }
      ]
    }

    HTTP/1.1 200 OK
    Content-Length: XXX
    Content-Type: application/alto-cdnifci+json
    {
      "meta" : {
        "vtag": {
          "resource-id": "my-default-cdnifci",
          "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
        }
      },
      "cdni-fci": {
        "capabilities": [
          {
            "capability-type": "FCI.DeliveryProtocol",
            "capability-value": {
              "delivery-protocols": [
                "http/1.1"
              ]
            },
            "footprints": [
              <Footprint objects>
            ]
          },
          {
            "capability-type": "FCI.DeliveryProtocol",
            "capability-value": {
              "delivery-protocols": [
                "https/1.1",
                "http/1.1"
              ]
            },
            "footprints": [
              <Footprint objects>
            ]
          }
        ]
```

```
      }
    }
```

5.7.3. Incremental Updates Example

In this example, the ALTO client only cares about the updates of one Delivery Protocol object whose value is "http/1.1". So it adds its limitation of capabilities in "input" field of the POST request.

```
  POST /updates/cdnifci HTTP/1.1
  Host: fcialtoupdate.example.com
  Accept: text/event-stream,application/alto-error+json
  Content-Type: application/alto-updatestreamparams+json
  Content-Length: ###

  { "add": {
      "my-fci-stream": {
          "resource-id": "my-filtered-cdnifci",
          "input": {
            "cdni-fci-capabilities": [
            {
              "capability-type": "FCI.DeliveryProtocol",
              "capability-value": {
                "delivery-protocols": [
                  "http/1.1"
                ]
              }
            }
          ]
        }
      }
    }
  }

  HTTP/1.1 200 OK
  Connection: keep-alive
  Content-Type: text/event-stream

  event: application/alto-updatestreamcontrol+json
  data: {"control-uri":
  data: "http://alto.example.com/updates/streams/3141592653590"}

  event: application/alto-cdnifci+json,my-fci-stream
  data: { ... full filtered CDNI FCI resource ... }

  event: application/merge-patch+json,my-fci-stream
  data: {
  data:   "meta": {
```

```
data:     "vtag": {
data:       "tag": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
data:     }
data:   },
data:   {
data:     "capability-type": "FCI.DeliveryProtocol",
data:     "capability-value": {
data:       "delivery-protocols": [
data:         "http/1.1"
data:       ]
data:     },
data:     "footprints": [
data:       <All footprint objects in http/1.1>
data:     ]
data:   }
data: }

event: application/json-patch+json,my-fci-stream
data: [
data:   {
data:     "op": "replace",
data:     "path": "/meta/vtag/tag",
data:     "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
data:   },
data:   { "op": "add",
data:     "path": "/cdni-fci/capabilities/0/footprints/-",
data:     "value": "ipv4:192.0.2.0/24"
data:   }
data: ]
```

6. Query Footprint Properties using ALTO Property Map Service

Besides retrieving footprints of given capabilities, another common requirement for uCDN is to query CDNI capabilities of given footprints.

Considering each footprint as an entity with properties including CDNI capabilities, the most natrual way to satisfy this requirement is to use the ALTO property map defined in [I-D.ietf-alto-unified-props-new]. In this section, we describe how ALTO clients look up properties for individual footprints. We firstly describe how to represent footprint objects as entities in the ALTO property map. And then we provide examples of the full property map and the filtered property map supporting CDNI capabilities, and their incremental updates.

6.1. Representing Footprint Objects as Unified Property Map Entities

A footprint object has two properties: footprint-type and footprint-value. A footprint-value is an array of footprint values conforming to the specification associated with the registered footprint type ("ipv4cidr", "ipv6cidr", "asn", and "countrycode"). Considering each ALTO entity defined in [I-D.ietf-alto-unified-props-new] also has two properties: entity domain type and domain-specific identifier, a straightforward approach to represent a footprint as an ALTO entity is to regard its footprint-type as an entity domain type, and its footprint value as a domain-specific identifier. According to [I-D.ietf-alto-unified-props-new], "ipv4" and "ipv6" are two predefined entity domain types, which can be used to represent "ipv4cidr" and "ipv6cidr" footprints respectively. However, no existing entity domain type can represent "asn" and "countrycode" footprints. To represent footprint-type "asn" and "countrycode", this document registers two new domains in Section 7 in addition to the ones in [I-D.ietf-alto-unified-props-new].

Here is an example of representing a footprint object as a set of entities in the ALTO property map.

{"footprint-type": "ipv4cidr", "footprint-value": ["192.0.2.0/24", "198.51.100.0/24"]} --> "ipv4:192.168.2.0/24", "ipv4:198.51.100.0/24"

6.1.1. ASN Domain

The ASN domain associates property values with Autonomous Systems in the Internet.

6.1.1.1. Entity Domain Type

asn

6.1.1.2. Domain-Specific Entity Identifiers

The entity identifiers of entities in an asn domain is encoded as a string consisting of the characters "as" (in lowercase) followed by the Autonomous System Number [RFC6793].

6.1.1.3. Hierarchy and Inheritance

There is no hierarchy or inheritance for properties associated with ASN.

6.1.2. COUNTRYCODE Domain

The COUNTRYCODE domain associates property values with countries.

6.1.2.1. Entity Domain Type

countrycode

6.1.2.2. Domain-Specific Entity Identifiers

The entity identifiers of entities in a countrycode domain is encoded as an ISO 3166-1 alpha-2 code [ISO3166-1] in lowercase.

6.1.2.3. Hierarchy and Inheritance

There is no hierarchy or inheritance for properties associated with country codes.

6.2. Examples

6.2.1. IRD Example

We use the same IRD example given by Section 3.7.1.

6.2.2. Property Map Example

This example shows a full property map in which entities are footprints and entities' property is "cdni-fci-capabilities".

```
GET /propmap/full/cdnifci HTTP/1.1
HOST: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json

HTTP/1.1 200 OK
Content-Length: ###
Content-Type: application/alto-propmap+json

{
  "property-map": {
    "meta": {
      "dependent-vtags": [
        {"resource-id": "my-default-cdnifci",
         "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"}
      ]
    },
    "countrycode:us": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}]
    },
    "ipv4:192.0.2.0/24": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}]
    },
    "ipv4:198.51.100.0/24": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}]
    },
    "ipv6:2001:db8::/32": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}]
    },
    "asn:as64496": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1",
                                                     "https/1.1"]}}]
    }
  }
}
```

6.2.3. Filtered Property Map Example

In this example, we use filtered property map service to get "pid" and "cdni-fci-capabilities" properties for two footprints "ipv4:192.0.2.0/24" and "ipv6:2001:db8::/32".

```
POST /propmap/lookup/cdnifci-pid HTTP/1.1
HOST: alto.example.com
Content-Type: application/alto-propmapparams+json
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length:

{
  "entities": [
    "ipv4:192.0.2.0/24",
    "ipv6:2001:db8::/32"
  ],
  "properties": [ "my-default-cdnifci.cdni-fci-capabilities",
                  "my-default-networkmap.pid" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: ###
Content-Type: application/alto-propmap+json

{
  "property-map": {
    "meta": {
      "dependent-vtags": [
         {"resource-id": "my-default-cdnifci",
           "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"},
         {"resource-id": "my-default-networkmap",
           "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf63"}
      ]
    },
    "ipv4:192.0.2.0/24": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}],
      "my-default-networkmap.pid": "pid1"
    },
    "ipv6:2001:db8::/32": {
      "my-default-cdnifci.cdni-fci-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}],
      "my-default-networkmap.pid": "pid3"
    }
  }
}
```

6.2.4. Incremental Updates Example

In this example, here is a client want to request updates for the properties "cdni-fci-capabilities" and "pid" for two footprints "ipv4:192.0.2.0/24" and "countrycode:fr".

```
POST /updates/properties HTTP/1.1
Host: alto.example.com
Accept: text/event-stream,application/alto-error+json
Content-Type: application/alto-updatestreamparams+json
Content-Length: ###

{ "add": {
  "property-map-including-capability-property": {
    "resource-id": "filtered-cdnifci-property-map",
      "input": {
        "properties": [ "my-default-cdnifci.cdni-fci-capabilities",
                        "my-default-networkmap.pid" ],
        "entities": [
```

```
            "ipv4:192.0.2.0/24",
            "ipv6:2001:db8::/32"
          ]
        }
      }
    }

    HTTP/1.1 200 OK
    Connection: keep-alive
    Content-Type: text/event-stream

    event: application/alto-updatestreamcontrol+json
    data: {"control-uri":
    data: "http://alto.example.com/updates/streams/1414213562373"}

    event: application/alto-cdnifci+json,my-fci-stream
    data: { ... full filtered unified property map ... }

    event: application/merge-patch+json,my-fci-stream
    data: {
    data:   "property-map":
    data:   {
    data:     "meta": {
    data:       "dependent-vtags": [
    data:         {"resource-id": "my-default-cdnifci",
    data:          "tag": "2beeac8ee23c3dd1e98a73fd30df80ece9fa5627"},
    data:         {"resource-id": "my-default-networkmap",
    data:          "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf63"}
    data:       ]
    data:     },
    data:     "ipv4:192.0.2.0/24":
    data:     {
    data:     "my-default-cdnifci.cdni-fci-capabilities": [
    data:       {"capability-type": "FCI.DeliveryProtocol",
    data:        "capability-value": {
    data:          "delivery-protocols": ["http/1.1"]}}]
    data:     }
    data:   }
    data: }

    event: application/json-patch+json,my-fci-stream
    data: {[
    data: {
    data:   { "op": "replace",
    data:     "path": "/meta/dependent-vtags/0/tag",
    data:     "value": "61b23185a50dc7b334577507e8f00ff8c3b409e4"
    data:   },
    data:   { "op": "replace",
```

```
data:     "path":
data:     "/property-map/countrycode:fr/my-default-networkmap.pid",
data:     "value": "pid5"
data:   }
data: }
data: ]}
```

7. IANA Considerations

7.1. CDNI Metadata Footprint Type Registry

Footprint Type	Description	Specification
altopid	A list of PID-names	RFCthis

Table 1: CDNI Metadata Footprint Type

[RFC Editor: Please replace RFCthis with the published RFC number for this document.]

7.2. ALTO Entity Domain Type Registry

As proposed in Section 11.2 of [I-D.ietf-alto-unified-props-new], "ALTO Entity Domain Type Registry" is requested. Besides, two new entity domain types are to be registered, listed in Table 2.

Identifier	Entity Address Encoding	Hierarchy & Inheritance
asn	See Section 6.1.1.2	None
countrycode	See Section 6.1.2.2	None

Table 2: ALTO Entity Domain Types

7.3. ALTO Entity Property Type Registry

As proposed in Section 11.3 of [I-D.ietf-alto-unified-props-new], "ALTO Entity Property Type Registry" is required. Besides, a new entity property type is to be registred, listed in Table 3.

Identifier	Intended Semantics
cdni-fci-capabilities	An array of CDNI FCI capability objects

Table 3: ALTO CDNI FCI Property Type

8. Security Considerations

As an extension of the base ALTO protocol [RFC7285], this document fits into the architecture of the base protocol, and hence the Security Considerations (Section 15) of the base protocol fully apply when this extension is provided by an ALTO server.

In the context of CDNI FCI, additional security considerations should be included as follows.

For authenticity and integrity of ALTO information, an attacker may disguise itself as an ALTO server for a dCDN, and provide false capabilities and footprints to a uCDN using the CDNI FCI map. Such false information may lead a uCDN to (1) select an incorrect dCDN to serve user requests or (2) skip uCDNs in good conditions.

For potential undesirable guidance from authenticated ALTO information, dCDNs can provide a uCDN with limited capabilities and smaller footprint coverage so that dCDNs can avoid transferring traffic for a uCDN which they should have to transfer.

For confidentiality and privacy of ALTO information, footprint properties integrated with ALTO unified property may expose network location identifiers (e.g., IP addresses or fine-grained PIDs).

For availability of ALTO services, an attacker may get the potential huge full CDNI FCI maps from an ALTO server in a dCDN continuously to run out of bandwidth resources of that ALTO server or may query filtered CDNI FCI services with complex capabilities to run out of computation resources of an ALTO server.

Protection strategies described in RFC 7285 can solve problems mentioned above well. However, the isolation of full/filtered CDNI FCI maps should also be considered.

If a dCDN signs agreements with multiple uCDNs, it must isolate full/filtered CDNI FCI maps for different uCDNs in that uCDNs will not redirect requests which should not have to served by this dCDN to this dCDN and it may not disclose extra information to uCDNs.

To avoid this risk, a dCDN may consider generating URIs of different full/filtered CDNI FCI maps by hashing its company ID, a uCDN's company ID as well as their agreements. And it needs to avoid expoing all full/filtered CDNI FCI maps resources in one of its IRDs.

9. Acknowledgments

The authors would like to thank Daryl Malas, Matt Caulfield for their timely reviews and invaluable comments.

Jan Seedorf is partially supported by the GreenICN project (GreenICN: Architecture and Applications of Green Information Centric Networking), a research project supported jointly by the European Commission under its 7th Framework Program (contract no. 608518) and the National Institute of Information and Communications Technology (NICT) in Japan (contract no. 167). 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 GreenICN project, the European Commission, or NICT.

10. References

10.1. Normative References

[ISO3166-1] The International Organization for Standardization, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", ISO 3166-1:2013, 2013.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <https://www.rfc-editor.org/info/rfc5693>.

[RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, DOI 10.17487/RFC6707, September 2012, <https://www.rfc-editor.org/info/rfc6707>.

[RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012, <https://www.rfc-editor.org/info/rfc6793>.

[RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <https://www.rfc-editor.org/info/rfc7285>.

[RFC8006] Niven-Jenkins, B., Murray, R., Caulfield, M., and K. Ma, "Content Delivery Network Interconnection (CDNI) Metadata", RFC 8006, DOI 10.17487/RFC8006, December 2016, <https://www.rfc-editor.org/info/rfc8006>.

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

10.2. Informative References

[I-D.ietf-alto-incr-update-sse]
Roome, W. and Y. Yang, "ALTO Incremental Updates Using Server-Sent Events (SSE)", draft-ietf-alto-incr-update-sse-17 (work in progress), July 2019.

[I-D.ietf-alto-path-vector]
Gao, K., Lee, Y., Randriamasy, S., Yang, Y., and J. Zhang, "ALTO Extension: Path Vector", draft-ietf-alto-path-vector-08 (work in progress), July 2019.

[I-D.ietf-alto-unified-props-new]
Roome, W., Randriamasy, S., Yang, Y., and J. Zhang, "Unified Properties for the ALTO Protocol", draft-ietf-alto-unified-props-new-08 (work in progress), July 2019.

[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.

Authors' Addresses

Jan Seedorf
HFT Stuttgart - Univ. of Applied Sciences
Schellingstrasse 24
Stuttgart 70174
Germany

Phone: +49-0711-8926-2801
Email: jan.seedorf@hft-stuttgart.de

Y.R. Yang
Tongji/Yale University
51 Prospect Street
New Haven, CT 06511
United States of America

Email: yry@cs.yale.edu
URI: http://www.cs.yale.edu/~yry/

Kevin J. Ma
Ericsson
43 Nagog Park
Acton, MA 01720
United States of America

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

Jon Peterson
NeuStar
1800 Sutter St Suite 570
Concord, CA 94520
United States of America

Email: jon.peterson@neustar.biz

Xiao Shawn Lin
Tongji University
4800 Cao'an Hwy
Shanghai 201804
China

Email: x.shawn.lin@gmail.com

Jingxuan Jensen Zhang
Tongji University
4800 Cao’an Hwy
Shanghai 201804
China

Email: jingxuan.zhang@tongji.edu.cn