ALTO & CDNI WGs J. Seedorf
Internet-Draft HFT Stuttgart - Univ. of Applied Sciences
Intended status: Standards Track Y. Yang
Expires: January 14, 2021 Tongji/Yale
K. Ma
Ericsson
J. Peterson
Neustar
J. Zhang
Tongji
July 13, 2020

Content Delivery Network Interconnection (CDNI) Request Routing: CDNI Footprint and Capabilities Advertisement using ALTO
draft-ietf-alto-cdni-request-routing-alto-12

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. This document defines an FCI protocol using the Application-Layer Traffic Optimization (ALTO) protocol, following the guidelines defined in [RFC8008].

Status of This Memo

Copyright Notice

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 the 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." At a high level, the scope of the CDNI Request Routing Interface, therefore, contains two main tasks: (1) determining if the dCDN (downstream CDN) is willing to accept a delegated content request, and (2) redirecting the content request coming from a uCDN (upstream CDN) to the proper entry point or entity in the dCDN.

Correspondingly, the request routing interface is broadly divided into two functionalities: (1) the CDNI Footprint & Capabilities Advertisement interface (FCI) defined in [RFC8008], and (2) the CDNI Request Routing Redirection interface (RI) defined in [RFC7975]. Since this document focuses on the first functionality (CDNI FCI), below is more details about it.

Specifically, CDNI FCI allows both an advertisement from a dCDN to a uCDN (push) and 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 defined 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 Advertisement Service".

There are multiple benefits in using ALTO as a transport protocol, as discussed 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 Advertisement Service", 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 new granularity to describe footprints. Section 5 introduces "Filtered CDNI Advertisement Service" to allow a uCDN to get footprints with given capabilities instead of getting the full resource, which can be large. Section 6 further shows another benefit of using ALTO: the ability to query footprint properties using ALTO unified properties. In this way, a uCDN can effectively fetch capabilities of footprints in which it is interested. IANA and security considerations are discussed in Section 7 and Section 8 respectively.

2. Background

The design of CDNI FCI transport using ALTO depends on the understanding of both FCI semantics and ALTO. Hence, this document starts with a non-normative review for both. The review uses the terminologies for CDNI as defined in [RFC6707], [RFC8006] and

[RFC8008]; those for ALTO as defined in [RFC7285] and [I-D.ietf-alto-unified-props-new].

2.1. Semantics of FCI Advertisement

[RFC8008] (CDNI "Footprint and Capabilities Semantics") defines the semantics of CDNI FCI, provides guidance on what Footprint and Capabilities mean in a CDNI context, and specifies the requirements on the CDNI FCI transport protocol. The definitions in [RFC8008] depend on [RFC8006]. Below is a non-normative review of key related points of [RFC8008] and [RFC8006]. For detailed information and normative specification, the reader is referred to these two RFCs.

- Multiple types of mandatory-to-implement 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 or 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.

- Multiple initial types of capabilities are defined in [RFC8008] including (1) Delivery Protocol, (2) Acquisition Protocol, (3) Redirection Mode, (4) Capabilities related to CDNI Logging, and (5) Capabilities related to CDNI Metadata. They are required in all cases and therefore considered as mandatory-to-implement capabilities for all CDNI FCI implementations.

- Footprint and capabilities are defined together and cannot be interpreted independently from each other. Specifically, [RFC8008] integrates footprint and capabilities with an approach of "capabilities with footprint restrictions", by expressing capabilities on a per footprint basis.

- Specifically, for all 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.

- o Given that a large part of Footprint and Capabilities Advertisement may 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 FCI. 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 (downstream CDN) to provide changes/updates regarding a footprint and/or capabilities that it has prior agreed to serve in a contract with a uCDN (upstream CDN). Hence, server push and incremental encoding will be necessary techniques.

2.2. ALTO Background and Benefits

Application-Layer Traffic Optimization (ALTO) [RFC7285] defines an approach for conveying network layer (topology) information to "guide" the resource provider selection process in distributed applications that can choose among several candidate resources providers to retrieve a given resource. Usually, it is assumed that an ALTO server conveys information that these applications cannot measure or have difficulty measuring themselves [RFC5693].

Originally, ALTO was motivated by optimizing cross-ISP traffic generated by P2P applications [RFC5693]. However, ALTO can also be used for improving the request routing in CDNs. In particular, the CDNI problem statement [RFC6707] explicitly mentions ALTO as a candidate protocol for "actual algorithms for selection of CDN or Surrogate by Request-Routing systems".

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

- o 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. This matches the need of CDNI: a uCDN wants to improve application layer CDN request routing by using information (provided by a dCDN) that the uCDN could not easily obtain otherwise. Hence, 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.

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

o Security: The identification between uCDNs and dCDNs is an important requirement. ALTO maps can be signed and hence provide inherent integrity protection. Please see Section 8.

o 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.

o Error-handling: The ALTO protocol provides extensive error-handling in the whole request and response process (see Section 8.5 of [RFC7285]). A CDNI FCI interface based on ALTO would inherit this this extensive error-handling framework. Please see Section 5.

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

o Server-initiated Notifications and Incremental Updates: When the footprint or the capabilities of a dCDN change (i.e., unexpectedly from the perspective of a uCDN), server-initiated notifications would enable a dCDN to inform a uCDN about such changes directly. 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 recent network and cost map from the dCDN, and therefore redirect requests to the 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].

o 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 a uCDN. This would enable a uCDN to determine whether a dCDN actually has the capabilities for a given 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 a uCDN for optimized dCDN 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 dCDN 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 Advertisement Service

The ALTO protocol is based on the ALTO Information Service Framework which consists of multiple 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 multiple initial services, e.g., the ALTO network map service and cost map service.

This document defines a new ALTO service called "CDNI Advertisement Service" which conveys JSON objects of media type "application/alto-cdni+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 Advertisement Service". Similar to other ALTO services, this document defines the ALTO information resource for the "CDNI Advertisement Service" as follows.

3.1. Media Type

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

3.2. HTTP Method

A CDNI Advertisement 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 Advertisement resource depends on other ALTO information resources. If the CDNI Advertisement resource has 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 Advertisement resource. See Section 4 for details of when and how to use it. Future documents may extend the CDNI Advertisement resource and allow other dependent resources.

3.6. Response

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

If a CDNI Advertisement response depends on other ALTO information resources, 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 Advertisement response is named "cdni-advertisement", which is a JSON object of type CDNIAdvertisementData:

```
    object {
      CDNIAdvertisementData cdni-advertisement;
    } InfoResourceCDNIAdvertisement : ResponseEntityBase;

    object {
      BaseAdvertisementObject capabilities-with-footprints<0..*>;
    } CDNIAdvertisementData;
```

Specifically, a CDNIAdvertisementData object is a JSON object that includes only one property named "capabilities-with-footprints", 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, below is a non-normative specification of BaseAdvertisementObject. 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-advertisement": {
        "capabilities-with-footprints": [
          {
            "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-advertisement": {
          "capabilities-with-footprints": [
            {
              "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.

This document only requires the ALTO server to provide the initial FCI-specific CDNI Payload Types defined in [RFC8008] as the mandatory-to-implement CDNI capabilities. There may be other documents extending BaseAdvertisementObject and additional CDNI capabilities. They are outside the scope of this document. To support them, future documents can extend the specification defined in 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 Advertisement resources), demonstrating a single, integrated environment.

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

HTTP/1.1 200 OK
Content-Length: 3571
Content-Type: application/alto-directory+json

{
  "meta" : {
    "default-alto-network-map": "my-default-network-map"
  },
  "resources": {
    "my-default-network-map": {
      "uri" : "https://alto.example.com/networkmap",
      "media-type" : "application/alto-networkmap+json"
    },
    "my-eu-netmap" : {
      "uri" : "https://alto.example.com/myeunetmap",
      "media-type" : "application/alto-networkmap+json"
    },
    "my-default-cdnifci": {
      "uri" : "https://alto.example.com/cdnifci",
      "media-type": "application/alto-cdni+json"
```

```
      },
      "my-cdnifci-with-pid-footprints": {
        "uri" : "https://alto.example.com/networkcdnifci",
        "media-type" : "application/alto-cdni+json",
        "uses" : [ "my-eu-netmap" ]
      },
      "my-filtered-cdnifci" : {
        "uri" : "https://alto.example.com/cdnifci/filtered",
        "media-type" : "application/alto-cdni+json",
        "accepts" : "application/alto-cdnifilter+json"
      },
      "cdnifci-property-map" : {
        "uri" : "https://alto.example.com/propmap/full/cdnifci",
        "media-type" : "application/alto-propmap+json",
        "uses": [ "my-default-cdni" ],
        "capabilities" : {
          "mappings": {
            "ipv4": [ "my-default-cdni.cdni-capabilities" ],
            "ipv6": [ "my-default-cdni.cdni-capabilities" ],
            "countrycode": [
              "my-default-cdni.cdni-capabilities" ],
            "asn": [ "my-default-cdni.cdni-capabilities" ]
          }
        }
      },
      "filtered-cdnifci-property-map" : {
        "uri" : "https://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-capabilities",
                      "my-default-network-map.pid" ],
            "ipv6": [ "my-default-cdni.cdni-capabilities",
                      "my-default-network-map.pid" ],
            "countrycode": [
              "my-default-cdni.cdni-capabilities" ],
            "asn": [ "my-default-cdni.cdni-capabilities" ]
          }
        }
      },
      "update-my-cdni-fci" : {
        "uri": "https:///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+json,application/json-patch+json",
             "my-cdnifci-with-pid-footprints" :
             "application/merge-patch+json,application/json-patch+json"
            }
          }
        },
        "update-my-props": {
          "uri" : "https://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

This basic example demonstrates a simple CDNI Advertisement resource, which does not depend on other resources. There are three BaseAdvertisementObjects in this resource 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-cdni+json,
           application/alto-error+json

   HTTP/1.1 200 OK
   Content-Length: 1235
   Content-Type: application/alto-cdni+json
```


```
   {
     "meta" : {
       "vtag": {
         "resource-id": "my-default-cdnifci",
         "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
       }
     },
     "cdni-advertisement": {
       "capabilities-with-footprints": [
         {
           "capability-type": "FCI.DeliveryProtocol",
           "capability-value": {
             "delivery-protocols": [
               "http/1.1"
             ]
           },
           "footprints": [
             {
               "footprint-type": "ipv4cidr",
               "footprint-value": [ "192.0.2.0/24" ]
             }
           ]
         },
         {
           "capability-type": "FCI.DeliveryProtocol",
           "capability-value": {
             "delivery-protocols": [
               "https/1.1",
               "http/1.1"
             ]
           },
           "footprints": [
             {
               "footprint-type": "ipv4cidr",
               "footprint-value": [ "198.51.100.0/24" ]
             }
           ]
         },
         {
           "capability-type": "FCI.AcquisitionProtocol",
```

```
            "capability-value": {
              "acquisition-protocols": [
                "https/1.1"
              ]
            },
            "footprints": [
              {
                "footprint-type": "ipv4cidr",
                "footprint-value": [ "203.0.113.0/24" ]
              }
            ]
          }
        ]
      }
    }
```


3.7.3. Incremental Updates Example

A benefit of using ALTO to provide CDNI Advertisement resources is that such resources 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 Advertisement response. Then when there is a change in the delivery-protocols in that http/1.1 is removed (from [http/1.1, https/1.1] to only https/1.1) due to maintenance of the https/1.1 clusters, the ALTO server regenerates the new CDNI Advertisement resource and pushes the full replacement to the ALTO client. Later on, the ALTO server notifies the ALTO client that "192.0.2.0/24" is added into the "ipv4" footprint object for delivery-protocol https/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: 92

{ "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: "https://alto.example.com/updates/streams/3141592653589"}

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

    event: application/alto-cdni+json,my-cdnifci-stream
    data: {
    data:   "meta": {
    data:     "vtag": {
    data:       "tag": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
    data:     }
    data:   },
    data:   "cdni-advertisement": {
    data:     "capabilities": [
    data:       {
    data:         "capability-type": "FCI.DeliveryProtocol",
    data:         "capability-value": {
    data:           "delivery-protocols": [
    data:             "https/1.1"
    data:           ]
    data:         },
    data:         "footprints": [
    data:           { "footprint-type": "ipv4cidr",
    data:             "footprint-value": [ "203.0.113.0/24" ]
    data:           }
    data:         ]
    data:       },
    data:       { ... other CDNI advertisement object ... }
    data:     ]
    data:   }
    data: }

    event: application/json-patch+json,my-cdnifci-stream
    data: [
    data:   { "op": "replace",
    data:     "path": "/meta/vtag/tag",
    data:     "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
    data:   },
    data:   { "op": "add",
    data:     "path": "/cdni-advertisement/capabilities-with-footprints
```

```
    /0/footprints/0/footprint-value/-",
    data:     "value": "192.0.2.0/24"
    data:   }
    data: ]
```

4. CDNI Advertisement 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).

To leverage this concept, this document defines a new CDNI Footprint Type called "altopid". A CDNI Advertisement resource can depend on an ALTO network map resource and use "altopid" footprints to compress its CDNI Footprint Payload.

Specifically, the "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 Advertisement resource using "altopid" footprints depends on a network map. For such a CDNI Advertisement resource, 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

The examples below use the same IRD given in Section 3.7.1.

4.2.2. ALTO Network Map for CDNI Advertisement Example

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

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

HTTP/1.1 200 OK
Content-Length: 309
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": [ "203.0.113.0/24" ]
    }
  }
}
```


4.2.3. ALTO PID Footprints in CDNI Advertisement

This example shows a CDNI Advertisement 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-cdni+json,application/alto-error+json

HTTP/1.1 200 OK
Content-Length: 738
Content-Type: application/alto-cdni+json
```


```
{
  "meta" : {
    "dependent-vtags" : [
      {
        "resource-id": "my-eu-netmap",
        "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
      }
    ]
  },
  "cdni-advertisement": {
    "capabilities-with-footprints": [
      { "capability-type": "FCI.DeliveryProtocol",
        "capability-value": [ "https/1.1" ],
        "footprints": [
          { "footprint-type": "altopid",
            "footprint-value": [ "south-france" ]
          }
        ]
      },
      { "capability-type": "FCI.AcquisitionProtocol",
        "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" and its dependent network map "my-eu-netmap". Considering two changes, the first one is to change footprints of the https/1.1 delivery protocol capability, and the second one is to remove "south-france" from the footprints of the https/1.1 acquisition 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: 183

{ "add": {
    "my-eu-netmap-stream": {
      "resource-id": "my-eu-netmap"
    },
    "my-netmap-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: "https://alto.example.com/updates/streams/3141592653590"}

event: application/alto-networkmap+json,my-eu-netmap-stream
data: { ... full Network Map of my-eu-netmap ... }

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

event: application/json-patch+json,my-netmap-cdnifci-stream
data: [
data:   { "op": "replace",
data:     "path": "/meta/vtag/tag",
data:     "value": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
data:   },
data:   { "op": "add",
data:     "path":
data:     "/cdni-advertisement/capabilities-with-footprints
/0/footprints/0/footprint-value/-",
data:     "value": "germany"
data:   }
data: ]

event: application/json-patch+json,my-netmap-cdnifci-stream
data: [
data:   { "op": "replace",
data:     "path": "/meta/vtag/tag",
```

```
data:     "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
data:   },
data:   { "op": "remove",
data:     "path":
data:     "/cdni-advertisement/capabilities-with-footprints
/1/footprints/0/footprint-value/1"
data:   }
data: ]
```


5. Filtered CDNI Advertisement using CDNI Capabilities

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

5.1. Media Type

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

5.2. HTTP Method

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

5.3. Accept Input Parameters

The input parameters for a filtered CDNI Advertisement 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-cdnifilter+json" which is a JSON object of type ReqFilteredCDNIAdvertisement, where:

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

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

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 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 Advertisement 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

Same to the "uses" field of the CDNI Advertisement resource (see Section 3.5).

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 Advertisement request is invalid if:

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

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

o 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 capability.

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

The returned CDNI Advertisement 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 Advertisement resource from which the filtered CDNI Advertisement 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

The examples below use the same IRD example as in Section 3.7.1.

5.7.2. Basic Example

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

```
POST /cdnifci/filtered HTTP/1.1
HOST: alto.example.com
Accept: application/alto-cdni+json
Content-Type: application/cdnifilter+json
Content-Length: 176
```

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


```
HTTP/1.1 200 OK
Content-Length: 571
Content-Type: application/alto-cdni+json
```


```
{
  "meta" : {
    "vtag": {
      "resource-id": "my-filtered-cdnifci",
      "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
    }
  },
  "cdni-advertisement": {
    "capabilities-with-footprints": [
      {
        "capability-type": "FCI.DeliveryProtocol",
        "capability-value": {
          "delivery-protocols": [
            "https/1.1",
            "http/1.1"
          ]
        },
        "footprints": [
          {
            "footprint-type": "ipv4cidr",
            "footprint-value": [ "198.51.100.0/24" ]
          }
        ]
      }
    ]
  }
}
```

5.7.3. Incremental Updates Example

In this example, the ALTO client only cares about the updates of one advertisement object for delivery protocol capability whose value includes "https/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: 346

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

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

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

event: application/alto-cdni+json,my-filtered-fci-stream
data: { ... filtered CDNI Advertisement resource ... }

event: application/json-patch+json,my-filtered-fci-stream
data: [
data:   {
data:     "op": "replace",
data:     "path": "/meta/vtag/tag",
```

```
     data:     "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
     data:   },
     data:   { "op": "add",
     data:     "/cdni-advertisement/capabilities-with-footprints
     /0/footprints/0/footprint-value/-",
     data:     "value": "192.0.2.0/24"
     data:   }
     data: ]
```


6. Query Footprint Properties using ALTO Property Map Service

Besides the requirement of 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, a natural way to satisfy this requirement is to use the ALTO property map as defined in [I-D.ietf-alto-unified-props-new]. This section describes how ALTO clients look up properties for individual footprints. First, it describes how to represent footprint objects as entities in the ALTO property map. Then it describes how to represent footprint capabilities as entity properties in the ALTO property map. Finally, it provides examples of the full property map and the filtered property map supporting CDNI capabilities, and their incremental updates.

6.1. Representing Footprint Objects as 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", "countrycode", and "altopid"). 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 represent its footprint-type as an entity domain type, and its footprint value as a domain-specific identifier.

Each existing footprint type can be represented as an entity domain type as follows:

o 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.

- "pid" is also a predefined entity domain type, which can be used to represent "altopid" footprints. Note that "pid" is a resource-specific entity domain. To represent an "altopid" footprint, the specifying information resource of the corresponding "pid" entity domain MUST be the dependent network map used by the CDNI Advertisement resource providing this "altopid" footprint.

- 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 of "ipv4cidr" type as a set of "ipv4" entities in the ALTO property map. The representation of the footprint object of "ipv6cidr" type is similar.

```
{ "footprint-type": "ipv4cidr",
  "footprint-value": ["192.0.2.0/24", "198.51.100.0/24"]
} --> "ipv4:192.0.2.0/24", "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 identifier of an entity 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 identifier of an entity 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. Representing CDNI Capabilities as Property Map Entity Properties

This document defines a new entity property type called "cdni-capabilities". An ALTO server can provide a property map resource mapping the "cdni-capabilities" entity property type for a CDNI Advertisement resource that it provides to an "ipv4", "ipv6", "asn" or "countrycode" entity domain.

6.2.1. Defining Information Resource Media Type for Property Type cdni-capabilities

The entity property type "cdni-capabilities" allows to define resource-specific entity properties. When resource-specific entity properties are defined with entity property type "cdni-capabilities", the defining information resource for a "cdni-capabilities" property MUST be a CDNI Advertisement resource provided by the ALTO server. The media type of the defining information resource for a "cdni-capabilities" property is therefore:

application/alto-cdni+json

6.2.2. Intended Semantics of Property Type cdni-capabilities

A "cdni-capabilities" property for an entity is to indicate all the CDNI capabilities that a corresponding CDNI Advertisement resource provides for the footprint represented by this entity. Thus, the value of a "cdni-capabilities" property MUST be a JSON array. Each element in a "cdni-capabilities" property MUST be an JSON object as format of CDNICapability (see Section 5.3). The value of a "cdni-capabilities" property for an "ipv4", "ipv6", "asn", "countrycode" or "altopid" entity MUST include all the CDNICapability objects that are provided by the defining CDNI Advertisement resource and the represented footprint object of this entity are in their footprint restrictions.

6.3. Examples

6.3.1. IRD Example

The examples use the same IRD example given by Section 3.7.1.

6.3.2. Property Map Example

This example shows a full property map in which entities are footprints and entities' property is "cdni-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: 1522
Content-Type: application/alto-propmap+json
```


```
{
  "property-map": {
    "meta": {
      "dependent-vtags": [
        { "resource-id": "my-default-cdnifci",
          "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"}
      ]
    },
    "countrycode:us": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["http/1.1"]}}]
    },
    "ipv4:192.0.2.0/24": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["http/1.1"]}}]
    },
    "ipv4:198.51.100.0/24": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["https/1.1", "http/1.1"]}}]
    },
    "ipv4:203.0.113.0/24": {
      "my-default-cdnifci.cdni-capabilities": [
```

```
            { "capability-type": "FCI.AcquisitionProtocol",
              "capability-value": {
                "acquisition-protocols": ["http/1.1"]}}]
        },
        "ipv6:2001:db8::/32": {
          "my-default-cdnifci.cdni-capabilities": [
            { "capability-type": "FCI.DeliveryProtocol",
              "capability-value": {
                "delivery-protocols": ["http/1.1"]}}]
        },
        "asn:as64496": {
          "my-default-cdnifci.cdni-capabilities": [
            { "capability-type": "FCI.DeliveryProtocol",
              "capability-value": {
                "delivery-protocols": ["https/1.1", "http/1.1"]}}]
        }
      }
    }
```


6.3.3. Filtered Property Map Example

This example uses the filtered property map service to get "pid" and "cdni-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: 181

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

HTTP/1.1 200 OK
Content-Length: 796
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-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-capabilities": [
        {"capability-type": "FCI.DeliveryProtocol",
         "capability-value": {"delivery-protocols": ["http/1.1"]}}],
      "my-default-networkmap.pid": "pid3"
    }
  }
}
```

6.3.4. Incremental Updates Example

In this example, the client is interested in updates for the properties "cdni-capabilities" and "pid" of 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: 337

{ "add": {
    "fci-propmap-stream": {
      "resource-id": "filtered-cdnifci-property-map",
      "input": {
        "properties": [ "my-default-cdnifci.cdni-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: "https://alto.example.com/updates/streams/1414213562373"}

event: application/alto-cdni+json,fci-propmap-stream
data: { ... filtered property map ... }

event: application/merge-patch+json,fci-propmap-stream
data: {
data:   "property-map": {
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:       "my-default-cdnifci.cdni-capabilities": [
data:         { "capability-type": "FCI.DeliveryProtocol",
data:           "capability-value": {
data:             "delivery-protocols": ["http/1.1", "https/1.1"]}}]
data:     }
data:   }
data: }

event: application/json-patch+json,fci-propmap-stream
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: ]
```


7. IANA Considerations

7.1. application/alto-* Media Types

This document registers two additional ALTO media types, listed in Table 1.

Type	Subtype	Specification
application	alto-cdni+json	Section 3
application	alto-cdnifilter+json	Section 5

Table 1: Additional ALTO Media Types.

Type name: application

Subtype name: This document registers multiple subtypes, as listed in Table 1.

Required parameters: n/a

Optional parameters: n/a

Encoding considerations: Encoding considerations are identical to those specified for the "application/json" media type. See [RFC7159].

Security considerations: Security considerations related to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285].

Interoperability considerations: This document specifies formats of conforming messages and the interpretation thereof.

Published specification: This document is the specification for these media types; see Table 1 for the section documenting each media type.

Applications that use this media type: ALTO servers and ALTO clients either stand alone or are embedded within other applications.

Additional information:

Magic number(s): n/a

File extension(s): This document uses the mime type to refer to protocol messages and thus does not require a file extension.

Macintosh file type code(s): n/a

Person & email address to contact for further information: See Authors' Addresses section.

Intended usage: COMMON

Restrictions on usage: n/a

Author: See Authors' Addresses section.

Change controller: Internet Engineering Task Force (mailto:iesg@ietf.org).

7.2. CDNI Metadata Footprint Type Registry

As proposed in Section 7.2 of [RFC8006], "CDNI Metadata Footprint Types" registry is requested. A new footprint type is to be registered, listed in Table 2.

Footprint Type	Description	Specification
altopid	A list of PID-names	Section 4 of RFCthis

Table 2: CDNI Metadata Footprint Type

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

7.3. 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. Two new entity domain types are to be registered, listed in Table 3.

Identifier	Entity Address Encoding	Hierarchy & Inheritance	Media Type of Defining Resource
asn	See Section 6.1.1.2 of RFCthis	None	None
countrycode	See Section 6.1.2.2 of RFCthis	None	None

Table 3: Additional ALTO Entity Domain Types

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

7.4. 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. A new entity property type is to be registered, listed in Table 4.

Identifier	Intended Semantics	Media Type of Defining Resource
cdni-capabilities	Section 6.2 of RFCthis	application/alto-cdni+json

Table 4: Additional ALTO Entity Property Type

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

8. Security Considerations

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

In the context of CDNI Advertisement, additional security considerations should be included as follows:

o 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 Advertisement service. Such false information may lead a uCDN to (1) select an incorrect dCDN to serve user requests, or (2) skip uCDNs in good conditions.

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

o 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).

o For availability of ALTO services, an attacker may conduct service degradation attacks using services defined in this document to disable ALTO services of a network. It may request potentially large, full CDNI Advertisement resources from an ALTO server in a dCDN continuously, to consume the bandwidth resources of that ALTO server. It may also query filtered property map services with many smaller individual footprints, to consume the computation resources of the ALTO server.

Although protection strategies as described in Section 15 of [RFC7285] should be applied to address aforementioned security considerations, one additional information leakage risk introduced by this document could not be addressed by these strategies. In particular, if a dCDN signs agreements with multiple uCDNs without any isolation, this dCDN may disclose extra information of one uCDN to another one. In that case, one uCDN may redirect requests which should not have to be served by this dCDN to it.

To reduce the risk, a dCDN should isolate full/filtered CDNI Advertisement resources for different uCDNs. It could consider generating URIs of different full/filtered CDNI Advertisement resources by hashing its company ID, a uCDN's company ID as well as their agreements. A dCDN should avoid exposing all full/filtered CDNI Advertisement resources in one of its IRDs.

9. Acknowledgments

The authors thank Matt Caulfield, Danny Alex Lachos Perez, Daryl Malas and Sanjay Mishra for their timely reviews and invaluable comments.

Jan Seedorf has been 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.

This document has also been supported by the Coordination Support Action entitled 'Supporting European Experts Presence in lnternational Standardisation Activities in ICT' ("StandlCT.eu") funded by the European Commission under the Horizon 2020 Programme with Grant Agreement no. 780439. 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 European Commission.

10. Contributors

Mr. Xiao Shawn Lin is an author of an early version of this document, with many contributions.

11. References

11.1. 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>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

11.2. 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-22 (work in progress), March 2020.

[I-D.ietf-alto-path-vector]
Gao, K., Randriamasy, S., Yang, Y., and J. Zhang, "ALTO Extension: Path Vector", draft-ietf-alto-path-vector-10 (work in progress), March 2020.

[I-D.ietf-alto-unified-props-new]
Roome, W., Randriamasy, S., Yang, Y., Zhang, J., and K. Gao, "Unified Properties for the ALTO Protocol", draft-ietf-alto-unified-props-new-11 (work in progress), March 2020.

[RFC7336] Peterson, L., Davie, B., and R. van Brandenburg, Ed., "Framework for Content Distribution Network Interconnection (CDNI)", RFC 7336, DOI 10.17487/RFC7336, August 2014, <https://www.rfc-editor.org/info/rfc7336>.

[RFC7975] Niven-Jenkins, B., Ed. and R. van Brandenburg, Ed., "Request Routing Redirection Interface for Content Delivery Network (CDN) Interconnection", RFC 7975, DOI 10.17487/RFC7975, October 2016, <https://www.rfc-editor.org/info/rfc7975>.

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
Yale University/PCL
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.ietf@gmail.com

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

Email: jon.peterson@neustar.biz

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

Email: jingxuan.zhang@tongji.edu.cn

ALTO
Internet-Draft
Intended status: Standards Track
Expires: 15 January 2021

K. Gao
Sichuan University
Y. Lee

S. Randriamasy
Nokia Bell Labs
Y.R. Yang
Yale University
J. Zhang
Tongji University
14 July 2020

ALTO Extension: Path Vector
draft-ietf-alto-path-vector-11

Abstract

This document is an extension to the base Application-Layer Traffic Optimization protocol [RFC7285]. The current ALTO Cost Services only allow applications to obtain cost values on an end-to-end path defined by its source and destination. The present extension provides abstracted information on particular network components or elements traversed by a path between its source and destination. Examples of such abstracted components are networks, data centers or links. This is useful for applications whose performance is impacted by particular network components they traverse or by their properties. Applications having the choice among several connection paths may use this information to select paths accordingly and improve their performance. In particular, they may infer that several paths share common links and prevent traffic bottlenecks by avoiding such paths. This document introduces a new cost type called Path Vector. A Path Vector is an array of entities that each identifies an abstracted representation of a network part and that are called Abstract Network Element (ANE). Each ANE is defined by a set of properties. ANE properties are conveyed by an ALTO information resource called "Property Map", that can be packed together with the Path Vectors in a multipart response. They can also be obtained via a separate ALTO request to a Property Map. An ALTO Property Map is an extension to the ALTO protocol, that is specified in another document entitled "Unified Properties for the ALTO Protocol" [I-D.ietf-alto-unified-props-new].

Status of This Memo

Table of Contents

1. Introduction

Network performance metrics are crucial to the Quality of Experience (QoE) of today's applications. The ALTO protocol allows Internet Service Providers (ISPs) to provide guidance, such as topological distance between different end hosts, to overlay applications. Thus, the overlay applications can potentially improve the QoE by better orchestrating their traffic to utilize the resources in the underlying network infrastructure.

Existing ALTO Cost Map and Endpoint Cost Service provide only cost information on an end-to-end path defined by its <source, destination> endpoints: The base protocol [RFC7285] allows the services to expose the topological distances of end-to-end paths, while various extensions have been proposed to extend the capability of these services, e.g., to express other performance metrics [I-D.ietf-alto-performance-metrics], to query multiple costs simultaneously [RFC8189], and to obtain the time-varying values [I-D.ietf-alto-cost-calendar].

While the existing extensions are sufficient for many overlay applications, however, the QoE of some overlay applications depends not only on the cost information of end-to-end paths, but also on some intermediate network components and their properties. For example, job completion time, which is an important QoE metric for a large-scale data analytics application, is impacted by shared bottlenecks inside the carrier network.

Predicting such information can be very complex without the help of the ISP [AAAI2019]. With proper guidance from the ISP, an overlay application may be able to schedule its traffic for better QoE. In the meantime, it may be helpful as well for ISPs if applications could avoid using bottlenecks or challenging the network with poorly scheduled traffic.

Despite the benefits, ISPs are not likely to expose details on their network paths: first for the sake of confidentiality, second because it may result in a huge volume and overhead, and last because it is difficult for ISPs to figure out what information and what details an application needs. Likewise, applications do not necessarily need all the network path details and are likely not able to understand them.

Therefore, it is beneficial for both parties if an ALTO server provides ALTO clients with an "abstract network state" that provides the necessary details to applications, while hiding the network complexity and confidential information. An "abstract network state" is a selected set of abstract representations of intermediate network components traversed by the paths between <source, destination> pairs combined with properties of these components that are relevant to the overlay applications' QoE. Both an application via its ALTO client and the ISP via the ALTO server can achieve better confidentiality and resource utilization by appropriately abstracting relevant path components. The pressure on the server scalability can also be reduced by abstracting components and their properties and combining them in a single response.

This document extends [RFC7285] to allow an ALTO server convey "abstract network state", for paths defined by their <source, destination> pairs. To this end, it introduces a new cost type called "Path Vector". A Path Vector is an array of identifiers of so-called Abstract Network Element (ANE). An ANE represents an abstract intermediate component traversed by a path. It can be associated with various properties. The associations between ANEs and their properties are encoded in an ALTO information resource called Unified Property Map, which is specified in [I-D.ietf-alto-unified-props-new].

For better confidentiality, this document aims to minimize information exposure. In particular, this document enables and recommends that first ANEs are constructed on demand, and second an ANE is only associated with properties that are requested by an ALTO client. A Path Vector response involved two ALTO Maps: the Cost Map that contains the Path Vector results and the up-to-date Unified Property Map that contains the properties requested for these ANEs. To enforce consistency and improve server scalability, this document uses the "multipart/related" message defined in [RFC2387] to return the two maps in a single response.

The rest of the document are organized as follows. Section 3 introduces the extra terminologies that are used in this document. Section 4 uses an illustrative example to introduce the additional requirements of the ALTO framework, and discusses potential use cases. Section 5 gives an overview of the protocol design. Section 6 and Section 7 specify the Path Vector extension to the ALTO IRD and the information resources, with some concrete examples presented in Section 8. Section 9 discusses the backward compatibility with the base protocol and existing extensions. Security and IANA considerations are discussed in Section 11 and Section 12 respectively.

2. Requirements Languages

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

When the words appear in lower case, they are to be interpreted with their natural language meanings.

3. Terminology

This document extends the ALTO base protocol [RFC7285] and the Unified Property Map extension [I-D.ietf-alto-unified-props-new]. In addition to the terms defined in these documents, this document also uses the following additional terms:

* Abstract Network Element (ANE): An Abstract Network Element is a representation of network components. It can be a link, a middlebox, a virtualized network function (VNF), etc., or their aggregations. An ANE can be constructed either statically in advance or on demand based on the requested information. In a response, each ANE is represented by a unique ANE Name. Note that an ALTO client must not assume ANEs in different responses but with the same ANE Name refer to the same network component(s).

* Path Vector: A Path Vector, or an ANE Path Vector, is a JSON array of ANE Names. It conveys the information that the path between a source and a destination traverses the ANEs in the same order as they appear in the Path Vector.

* Path Vector resource: A Path Vector resource refers to an ALTO resource which supports the extension defined in this document.

* Path Vector cost type: The Path Vector cost type is a special cost type, which is specified in Section 6.5. When this cost type is present in an IRD entry, it indicates that the information resource is a Path Vector resource. When this cost type is present in a Cost Map or an Endpoint Cost Map, it indicates each cost value must be interpreted as a Path Vector.

* Path Vector request: A Path Vector request refers to the POST message sent to an ALTO Path Vector resource.

* Path Vector response: A Path Vector response refers to the multipart/related message returned by a Path Vector resource.

4. Problem Statement

4.1. Design Requirements

This section gives an illustrative example of how an overlay application can benefit from the Path Vector extension.

Assume that an application has control over a set of flows, which may go through shared links or switches and share a bottleneck. The application hopes to schedule the traffic among multiple flows to get better performance. The capacity region information for those flows will benefit the scheduling. However, existing cost maps can not reveal such information.

Specifically, consider a network as shown in Figure 1. The network has 7 switches (sw1 to sw7) forming a dumb-bell topology. Switches sw1/sw3 provide access on one side, sw2/sw4 provide access on the other side, and sw5-sw7 form the backbone. Endhosts eh1 to eh4 are connected to access switches sw1 to sw4 respectively. Assume that the bandwidth of link eh1 -> sw1 and link sw1 -> sw5 are 150 Mbps, and the bandwidth of the rest links are 100 Mbps.

```
                                 +------+
                                 |      |
                               --+ sw6  +--
                             /   |      |  \
       PID1 +-----+         /    +------+   \          +-----+  PID2
       eh1__|     |_       /                 \     ____|     |__eh2
   1.2.3.4  | sw1 | \   +--|---+         +---|--+ /    | sw2 |  2.3.4.5
            +-----+  \  |      |         |      |/     +-----+
                      \_| sw5  +---------+ sw7  |
       PID3 +-----+   / |      |         |      |\     +-----+  PID4
       eh3__|     |__/  +------+         +------+ \____|     |__eh4
   3.4.5.6  | sw3 |                                    | sw4 |  4.5.6.7
            +-----+                                    +-----+
```

Figure 1: Raw Network Topology

The single-node ALTO topology abstraction of the network is shown in Figure 2.

```
                    +----------------------+
          {eh1}     |                      |     {eh2}
          PID1      |                      |     PID2
            +-------+                      +-------+
                    |                      |
                    |                      |
          {eh3}     |                      |     {eh4}
          PID3      |                      |     PID4
            +-------+                      +-------+
                    |                      |
                    +----------------------+
```

Figure 2: Base Single-Node Topology Abstraction

Consider an application overlay (e.g., a large-scale data analytics system) which wants to optimize the total throughput of the traffic among a set of end host <source, destination> pairs, say eh1 -> eh2 and eh1 -> eh4. The application can request a cost map providing end-to-end available bandwidth, using "availbw" as cost-metric and "numerical" as cost-mode.

The application will receive from the ALTO server that the bandwidth of eh1 -> eh2 and eh1 -> eh4 are both 100 Mbps. But this information is not enough to determine the optimal total throughput. Consider the following two cases:

* Case 1: If eh1 -> eh2 uses the path eh1 -> sw1 -> sw5 -> sw6 -> sw7 -> sw2 -> eh2 and eh1 -> eh4 uses path eh1 -> sw1 -> sw5 -> sw7 -> sw4 -> eh4, then the application will obtain 150 Mbps at most.

* Case 2: If eh1 -> eh2 uses the path eh1 -> sw1 -> sw5 -> sw7 -> sw2 -> eh2 and eh1 -> eh4 uses the path eh1 -> sw1 -> sw5 -> sw7 -> sw4 -> eh4, then the application will obtain only 100 Mbps at most.

To allow applications to distinguish the two aforementioned cases, the network needs to provide more details. In particular:

* For eh1 -> eh2, the ALTO server must give more details which is critical for the overlay application to distinguish between Case 1 and Case 2 and to compute the optimal total throughput accordingly.

* The ALTO server must allow the client to distinguish the common network components shared by eh1 -> eh2 and eh1 -> eh4, e.g., eh1 - sw1 and sw1 - sw5 in Case 1.

* The ALTO server must give details on the properties of the network components used by eh1 -> eh2 and eh1 -> eh4, e.g., the available bandwidth between eh1 - sw1, sw1 - sw5, sw5 - sw7, sw5 - sw6, sw6 - sw7, sw7 - sw2, sw7 - sw4, sw2 - eh2, sw4 - eh4 in Case 1.

In general, we can conclude that to support the multiple flow scheduling use case, the ALTO framework must be extended to satisfy the following additional requirements:

AR1: An ALTO server must provide essential information on intermediate network components on the path of a <source, destination> pair that are critical to the QoE of the overlay application.

AR2: An ALTO server must provide essential information on how the paths of different <source, destination> pairs share a common network component.

AR3: An ALTO server must provide essential information on the properties associated to the network components.

The Path Vector extension defined in this document propose a solution to provide these details.

4.2. Recent Use Cases

While the multiple flow scheduling problem is used to help identify the additional requirements, the Path Vector extension can be applied to a wide range of applications. This section highlights some real use cases that are recently reported. See [I-D.bernstein-alto-topo] for a more comprehensive survey of use cases where extended network topology information is needed.

4.2.1. Large-scale Data Analytics

One potential use case of the Path Vector extension is for large-scale data analytics such as [SENSE] and [LHC], where data of Gigabytes, Terabytes and even Petabytes are transferred. For these applications, the QoE is usually measured as the job completion time, which is related to the completion time of the slowest data transfer. With the Path Vector extension, an ALTO client can identify bottlenecks inside the network. Therefore, the overlay application can make optimal traffic distribution or resource reservation (i.e., proportional to the size of the transferred data), leading to optimal job completion time and network resource utilization.

4.2.2. Context-aware Data Transfer

It is sometimes important to know how the capabilities of various network components between two end hosts, especially in the mobile environment. With the Path Vector extension, an ALTO client may query the "network context" information, i.e., whether the two hosts are connected to the access network through a wireless link or a wire, and the capabilities of the access network. Thus, the client may use different data transfer mechanisms, or even deploy different 5G User Plane Functions (UPF) [I-D.ietf-dmm-5g-uplane-analysis] to optimize the data transfer.

4.2.3. CDN and Service Edge

A growing trend in today's applications is to bring storage and computation closer to the end user for better QoE, such as Content Delivery Network (CDN), AR/VR, and cloud gaming, as reported in various recent documents ([I-D.contreras-alto-service-edge], [I-D.huang-alto-mowie-for-network-aware-app], and [I-D.yang-alto-deliver-functions-over-networks]).

With the Path Vector extension, an ALTO server can selectively reveal the CDNs and service edges that reside along the paths between different end hosts, together with their properties such as available Service Level Agreement (SLA) plans. Otherwise, the ALTO client may have to make multiple queries and potentially with the complete list of CDNs and/or service edges. While both approaches offer the same information, making multiple queries introduces larger delay and more overhead on both the ALTO server and the ALTO client.

5. Path Vector Extension: Overview

This section gives a non-normative overview of the Path Vector extension. It is assumed that readers are familiar with both the base protocol [RFC7285] and the Unified Property Map extension [I-D.ietf-alto-unified-props-new].

To satisfies the additional requirements, this extension:

1. introduces Abstract Network Element (ANE) as the abstraction of intermediate network components,

2. extends the Cost Map and Endpoint Cost Service to convey the intermediate network components traversed by the path of a <source, destination> pair as Path Vectors,

3. uses the Unified Property Map to convey the association between the intermediate network components and their properties.

Thus, an ALTO client can learn about the intermediate network components that are critical to the QoE of a <source, destination> pair by investigating the corresponding Path Vector value (AR1), identify common network components if an ANE appears in the Path Vectors of multiple <source, destination> pairs (AR2), and retrieve the properties of the network components by searching the Unified Property Map (AR3).

5.1. Abstract Network Element

This extension introduces Abstract Network Element (ANE) as an indirect and network-agnostic way to specify an aggregation of intermediate network components between a source and a destination. Specifically, an ANE is a string of type ANEName as specified in Section 6.1 and its associated set of properties.

5.1.1. ANE Domain

In this extension, the associations between ANE and the properties are conveyed in a Unified Property Map. Thus, they must follow the mechanisms specified in the [I-D.ietf-alto-unified-props-new].

Specifically, this document defines a new entity domain called "ane" as specified in Section 5.1.1 and defines two initial properties for the "ane" domain.

5.1.2. Ephemeral ANE and Persistent ANE

For different requests, there can be different ways of grouping network components and assigning ANEs. For example, an ALTO server may define an ANE for each aggregated bottleneck link between the sources and destinations specified in the request. As the aggregated bottleneck links vary for different combinations of sources and destinations, the ANEs are ephemeral and are no longer valid after the request completes. Thus, the scope of ephemeral ANEs are limited to the corresponding Path Vector response.

While ephemeral ANEs returned by a Path Vector response do not exist beyond that response, some of them may represent entities that are persistent and defined in a standalone Property Map. Indeed, it may be useful for clients to occasionally query properties on persistent entities, without caring about the path that traverses them. Persistent entities have a persistent ID that is registered in a Property Map, together with their properties.

5.1.3. Property Filtering

Resource-constrained ALTO clients may benefit from the filtering of Path Vector query results at the ALTO server, as an ALTO client may only require a subset of the available properties.

Specifically, the available properties for a given resource are announced in the Information Resource Directory as a new capability called "ane-property-names". The selected properties are specified in a filter called "ane-property-names" in the request body, and the response must return and only return the selected properties for the ANEs in the response.

The "ane-property-names" capability for Cost Map and for Endpoint Cost Service are specified in Section 7.1.4 and Section 7.2.4 respectively. The "ane-property-names" filter for Cost Map and Endpoint Cost Service are specified in Section 7.1.3 and Section 7.2.3 accordingly.

5.2. Path Vector Cost Type

For an ALTO client to correctly interpret the Path Vector, this extension specifies a new cost type called the Path Vector cost type, which must be included both in the Information Resource Directory and the ALTO Cost Map or Endpoint Cost Map so that an ALTO client can correct interpret the cost values.

The Path Vector cost type must convey both the interpretation and semantics in the "cost-mode" and "cost-metric" respectively. Unfortunately, a single "cost-mode" value cannot fully specify the interpretation of a Path Vector, which is a compound data type. For example, in programming languages such as Java, a Path Vector will have the type of JSONArray[ANEName].

Instead of extending the "type system" of ALTO, this document takes a simple and backward compatible approach. Specifically, the "cost-mode" of the Path Vector cost type is "array", which indicates the value is a JSON array. Then, an ALTO client must check the value of the "cost-metric". If the value is "ane-path", meaning the JSON array should be further interpreted as a path of ANENames.

The Path Vector cost type is specified in Section 6.5.

5.3. Multipart Path Vector Response

For a basic ALTO information resource, a response contains only one type of ALTO resources, e.g., Network Map, Cost Map, or Property Map. Thus, only one round of communication is required: An ALTO client sends a request to an ALTO server, and the ALTO server returns a response, as shown in Figure 3.

```
         ALTO client                              ALTO server
              |-------------- Request ---------------->|
              |<------------- Response ----------------|
```

Figure 3: A Typical ALTO Request and Response

The Path Vector extension, on the other hand, involves two types of information resources: Path Vectors conveyed in a Cost Map or an Endpoint Cost Map, and ANE properties conveyed in a Unified Property Map. Instead of two consecutive message exchanges, the Path Vector extension enforces one round of communication. Specifically, the Path Vector extension requires the ALTO client to include the source and destination pairs and the requested ANE properties in a single request, and encapsulates both Path Vectors and properties associated with the ANEs in a single response, as shown in Figure 4.

```
         ALTO client                              ALTO server
              |------------- PV Request -------------->|
              |<----- PV Response (Cost Map Part) -----|
              |<--- PV Response (Property Map Part) ---|
```

Figure 4: The Path Vector Extension Request and Response

This design is based on the following considerations:

1. Since ANEs may be constructed on demand, and potentially based on the requested properties (See Section 5.1 for more details). If sources and destinations are not in the same request as the properties, an ALTO server either cannot construct ANEs on-demand, or must wait until both requests are received.

2. As ANEs may be constructed on demand, mappings of each ANE to its underlying network devices and resources can be specific to the request. In order to respond to the Property Map request correctly, an ALTO server must store the mapping of each Path Vector request until the client fully retrieves the property information. The "stateful" behavior may substantially harm the server scalability and potentially lead to Denial-of-Service attacks.

One approach to realize the one-round communication is to define a new media type to contain both objects, but this violates modular design. This document follows the standard-conforming usage of "multipart/related" media type defined in [RFC2387] to elegantly combine the objects. Path Vectors are encoded as a Cost Map or an Endpoint Cost Map, and the Property Map is encoded as a Unified Propert Map. They are encapsulated as parts of a multipart message. The modular composition allows ALTO servers and clients to reuse the data models of the existing information resources. Specifically, this document addresses the following practical issues using "multipart/related".

5.3.1. Identifying the Media Type of the Root Object

ALTO uses media type to indicate the type of an entry in the Information Resource Directory (IRD) (e.g., "application/alto-costmap+json" for Cost Map and "application/alto-endpointcost+json" for Endpoint Cost Map). Simply putting "multipart/related" as the media type, however, makes it impossible for an ALTO client to identify the type of service provided by related entries.

To address this issue, this document uses the "type" parameter to indicate the root object of a multipart/related message. For a Cost Map resource, the "media-type" in the IRD entry must be "multipart/related" with the parameter "type=application/alto-costmap+json"; for an Endpoint Cost Service, the parameter must be "type=application/alto-endpointcost+json".

5.3.2. References to Part Messages

The ALTO SSE extension (see [I-D.ietf-alto-incr-update-sse]) uses "client-id" to demultiplex push updates. However, "client-id" is provided for each request, which introduces ambiguity when applying SSE to a Path Vector resource.

To address this issue, an ALTO server must assign a unique identifier to each part of the "multipart/related" response message. This identifier, referred to as a Part Resource ID (See Section 6.6 for details), must be present in the part message's "Resource-Id" header. The MIME part header must also contain the "Content-Type" header, whose value is the media type of the part (e.g., "application/alto-costmap+json", "application/alto-endpointcost+json", or "application/alto-propmap+json").

If an ALTO server provides incremental updates for this Path Vector resource, it must generate incremental updates for each part separately. The client-id must have the following format:

pv-client-id '.' part-resource-id

where pv-client-id is the client-id assigned to the Path Vector request, and part-resource-id is the "Resource-Id" header value of the part. The media-type must match the "Content-Type" of the part.

The same problem applies to the part messages as well. The two parts must contain a version tag, which SHOULD contain a unique Resource ID. This document requires the resource-id in a Version Tag to have the following format:

pv-resource-id '.' part-resource-id

where pv-resource-id is the resource ID of the Path Vector resource in the IRD entry, and the part-resource-id has the same value as the "Resource-Id" header of the part.

5.3.3. Order of Part Messages

According to [RFC2387], the Path Vector part, whose media type is the same as the "type" parameter of the multipart response message, is the root object. Thus, it is the element the application processes first. Even though the "start" parameter allows it to be placed anywhere in the part sequence, it is RECOMMENDED that the parts arrive in the same order as they are processed, i.e., the Path Vector part is always put as the first part, followed by the property map part. It is also RECOMMENDED that when doing so, an ALTO server SHOULD NOT set the "start" parameter, which implies the first part is the root object.

6. Specification: Basic Data Types

6.1. ANE Name

An ANE Name is encoded as a JSON string with the same format as that of the type PIDName (Section 10.1 of [RFC7285]).

The type ANEName is used in this document to indicate a string of this format.

6.2. ANE Domain

The ANE domain associates property values with the Abstract Network Elements in a Property Map. Accordingly, the ANE domain always depends on a Property Map.

6.2.1. Entity Domain Type

ane

6.2.2. Entity Identifier Encoding

The entity identifier of the "ane" domain has the same format as defined in Section 5.1.3 in [I-D.ietf-alto-unified-props-new], and the DomainTypeSpecificEntityID part has the same format as the ANEName type.

6.2.3. Hierarchy and Inheritance

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

6.2.4. Media Type of Defining Resource

When resource specific domains are defined with entities of domain type "ane", the defining resource for entity domain type "ane" MUST be a Property Map. The media type of defining resources for the "ane" domain is:

application/alto-propmap+json

Specifically, the defining resource of ephemeral ANEs is the Property Map part of the multipart response. The defining resource of persistent ANEs is the Property Map on which standalone queries for properties of persistent ANEs are made.

6.3. ANE Property Name

An ANE Property Name is encoded as a JSON string with the same format as that of Entity Property Name (Section 5.2.2 of [I-D.ietf-alto-unified-props-new]).

6.4. Initial ANE Property Types

In this document, two initial ANE property types are specified, "max-reservable-bandwidth" and "persistent-entity-id".

Note that the two property types defined in this document do not depend on any information resource, so their ResourceID part must be empty.

```
                                        ----- L1
                                       /
           PID1   +---------------+ 10 Gbps +----------+    PID3
    1.2.3.0/24+---+ +-----------+ +---------+          +---+3.4.5.0/24
                  | |   MEC1    | |         |          |
                  | +-----------+ |   +-----+          |
           PID2   |               |   |     +----------+
    2.3.4.0/24+---+               |   |         NET3
                  |               |   | 15 Gbps
                  |               |   |        \
                  +---------------+   |         -------- L2
                        NET1          |
                               +---------------+
                               | +-----------+ |   PID4
                               | |   MEC2    | +---+4.5.6.0/24
                               | +-----------+ |
                               +---------------+
                                     NET2
```

Figure 5: Examples of ANE Properties

In this document, Figure 5 is used to illustrate the use of the two initial ANE property types. There are 3 sub-networks (NET1, NET2 and NET3) and two interconnection links (L1 and L2). It is assumed that each sub-network has sufficiently large bandwidth to be reserved.

6.4.1. New ANE Property Type: Maximum Reservable Bandwidth

Identifier: "max-reservable-bandwidth"

Intended Semantics: The maximum reservable bandwidth property stands for the maximum bandwidth that can be reserved for all the traffic that traverses an ANE. The value MUST be encoded as a non-negative numerical cost value as defined in Section 6.1.2.1 of [RFC7285] and the unit is bit per second. If this property is requested but not present in an ANE, it MUST be interpreted as that the ANE does not support bandwidth reservation.

Security Considerations: ALTO entity properties expose information to ALTO clients. ALTO service providers should be made aware of the security ramifications related to the exposure of an entity property.

To illustrate the use of "max-reservable-bandwidth", consider the network in Figure 5. An ALTO server can create an ANE for each interconnection link, where the initial value for "max-reservable-bandwidth" is the link capacity.

6.4.2. New ANE Property Type: Persistent Entity ID

Identifier: "persistent-entity-id"

Intended Semantics: The persistent entity ID property is the entity identifier of the persistent ANE associated with an ephemeral ANE. The value of this property is encoded with the format defined in Section 5.1.3 of [I-D.ietf-alto-unified-props-new]. In this format, the entity ID combines:

* a defining information resource for the ANE on which a "persistent-entity-id" is queried, which is the property map defining the ANE as a persistent entity, together with the properties

* the persistent name of the ANE in this property map

With this format, the client has all the needed information for further standalone query properties on the persistent ANE.

Security Considerations: ALTO entity properties expose information to ALTO clients. ALTO service providers should be made aware of the security ramifications related to the exposure of an entity property.

To illustrate the use of "persistent-entity-id", consider the network in Figure 5. Assume the ALTO server has a Property Map resource called "mec-props" that defines persistent ANEs "MEC1" and "MEC2" that represent the corresponding mobile edge computing (MEC) clusters. The "persistent-entity-id" of the ephemeral ANE that is associated with MEC1 has the value "mec-props.ane:MEC1".

6.5. Path Vector Cost Type

This document defines a new cost type, which is referred to as the "Path Vector" cost type. An ALTO server MUST offer this cost type if it supports the Path Vector extension.

6.5.1. Cost Metric: ane-path

The cost metric "ane-path" indicates the value of such a cost type conveys an array of ANE names, where each ANE name uniquely represents an ANE traversed by traffic from a source to a destination.

6.5.2. Cost Mode: array

The cost mode "array" indicates that every cost value in a Cost Map or an Endpoint Cost Map MUST be interpreted as a JSON array object.

Note that this cost mode only requires the cost value to be a JSON array of JSONValue. However, an ALTO server that enables this extension MUST return a JSON array of ANEName (Section 6.1) when the cost metric is "ane-path".

6.6. Part Resource ID

A Part Resource ID is encoded as a JSON string with the same format as that of the type ResourceID (Section 10.2 of [RFC7285]).

NOTE: Even though the client-id assigned to a Path Vector request and the Part Resource ID may contain up to 64 characters by their own definition, their concatenation (see Section 5.3.2) MUST also conform to the same length constraint. The same requirement applies to the resource ID of the Path Vector resource, too. Thus, it is RECOMMENDED to limit the length of resource ID and client ID related to a Path Vector resource to 31 characters.

7. Specification: Service Extensions

7.1. Multipart Filtered Cost Map for Path Vector

This document introduces a new ALTO resource called multipart filtered cost map resource, which allows an ALTO server to provide other ALTO resources associated to the cost map resource in the same response.

7.1.1. Media Type

The media type of the multipart filtered cost map resource is "multipart/related;type=application/alto-costmap+json".

7.1.2. HTTP Method

The multipart filtered cost map is requested using the HTTP POST method.

7.1.3. Accept Input Parameters

The input parameters of the multipart filtered cost map are supplied in the body of an HTTP POST request. This document extends the input parameters to a filtered cost map with a data format indicated by the media type "application/alto-costmapfilter+json", which is a JSON object of type PVReqFilteredCostMap, where:

```
object {
  [EntityPropertyName ane-property-names<0..*>;]
} PVReqFilteredCostMap : ReqFilteredCostMap;
```

with fields:

ane-property-names: A list of properties that are associated with the ANEs. Each property in this list MUST match one of the supported ANE properties indicated in the resource's "ane-property-names" capability. If the field is NOT present, it MUST be interpreted as an empty list, indicating that the ALTO server MUST NOT return any property in the Unified Property part.

Example: Consider the network in Figure 1. If an ALTO client wants to query the "max-reservable-bandwidth" between PID1 and PID2, it can submit the following request.

```
POST /costmap/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;type=application/alto-costmap+json,
        application/alto-error+json
Content-Length: [TBD]
Content-Type: application/alto-costmapfilter+json

{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "pids": {
    "srcs": [ "PID1" ],
    "dsts": [ "PID2" ]
  },
  "ane-property-names": [ "max-reservable-bandwidth" ]
}
```

7.1.4. Capabilities

The multipart filtered cost map resource extends the capabilities defined in Section 11.3.2.4 of [RFC7285]. The capabilities are defined by a JSON object of type PVFilteredCostMapCapabilities:

```
object {
  [EntityPropertyName ane-property-names<0..*>;]
} PVFilteredCostMapCapabilities : FilteredCostMapCapabilities;
```

with fields:

cost-type-names: The "cost-type-names" field MUST only include the Path Vector cost type, unless explicitly documented by a future extension. This also implies that the Path Vector cost type MUST be defined in the "cost-types" of the Information Resource Directory’s "meta" field.

cost-constraints: If the "cost-type-names" field includes the Path Vector cost type, "cost-constraints" field MUST be "false" or not present unless specifically instructed by a future document.

testable-cost-type-names: If the "cost-type-names" field includes the Path Vector cost type, the Path Vector cost type MUST NOT be included in the "testable-cost-type-names" field unless specifically instructed by a future document.

ane-property-names: Defines a list of ANE properties that can be returned. If the field is NOT present, it MUST be interpreted as an empty list, indicating the ALTO server cannot provide any ANE property.

7.1.5. Uses

This member MUST include the resource ID of the network map based on which the PIDs are defined. If this resource supports "persistent-entity-id", it MUST also include the defining resources of persistent ANEs that may appear in the response.

7.1.6. Response

The response MUST indicate an error, using ALTO protocol error handling, as defined in Section 8.5 of [RFC7285], if the request does no.

The "Content-Type" header of the response MUST be "multipart/related" as defined by [RFC2387] with the following parameters:

type: The type parameter MUST be "application/alto-costmap+json". Note that [RFC2387] permits both parameters with and without the double quotes.

start: The start parameter is as defined in [RFC2387]. If present, it MUST have the same value as the "Resource-Id" header of the Path Vector part.

boundary: The boundary parameter is as defined in [RFC2387].

The body of the response consists of two parts:

* The Path Vector part MUST include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" MUST has the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-costmap+json".

  The body of the Path Vector part MUST be a JSON object with the same format as defined in Section 11.2.3.6 of [RFC7285]. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned cost map. The resource ID of the version tag MUST follow the format in Section 5.3.2. The "meta" field MUST also include the "dependent-vtags" field, whose value is a single-element array to indicate the version tag of the network map used, where the network map is specified in the "uses" attribute of the multipart filtered cost map resource in IRD.

* The Unified Property Map part MUST also include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" has the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-propmap+json".

  The body of the Unified Property Map part MUST be a JSON object with the same format as defined in Section 4.6 of [I-D.ietf-alto-unified-props-new]. The JSON object MUST include the "dependent-vtags" field in the "meta" field. The value of the "dependent-vtags" field MUST be an array of VersionTag objects as defined by Section 10.3 of [RFC7285]. The "vtag" of the Path Vector part MUST be included in the "dependent-vtags". If "persistent-entity-id" is requested, the version tags of the dependent resources that may expose the entities in the response MUST also be included. The PropertyMapData has one member for each ANEName that appears in the Path Vector part, which is an entity identifier belonging to the self-defined entity domain as defined in Section 5.1.2.3 of [I-D.ietf-alto-unified-props-new]. The EntityProps has one member for each property requested by an ALTO client if applicable.

If the "start" parameter is not present, the Path Vector part MUST be the first part in the multipart response.

Example: Consider the network in Figure 1. The response of the example request in Section 7.1.3 is as follows, where "ANE1" represents the aggregation of all the switches in the network.

```
HTTP/1.1 200 OK
Content-Length: [TBD]
Content-Type: multipart/related; boundary=example-1;
              type=application/alto-costmap+json

--example-1
Resource-Id: costmap
Content-Type: application/alto-costmap+json

{
  "meta": {
    "vtag": {
      "resource-id": "filtered-cost-map-pv.costmap",
      "tag": "d827f484cb66ce6df6b5077cb8562b0a"
    },
    "dependent-vtags": [
      {
        "resource-id": "my-default-networkmap",
        "tag": "75ed013b3cb58f896e839582504f6228"
      }
    ],
    "cost-type": { "cost-mode": "array", "cost-metric": "ane-path" }
  },
  "cost-map": {
    "PID1": { "PID2": ["ANE1"] }
  }
}
--example-1
Resource-Id: propmap
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "filtered-cost-map-pv.costmap",
        "tag": "d827f484cb66ce6df6b5077cb8562b0a"
      }
    ]
  },
  "property-map": {
    ".ane:ANE1": { "max-reservable-bandwidth": 100000000 }
  }
}
```

7.2. Multipart Endpoint Cost Service for Path Vector

This document introduces a new ALTO resource called multipart endpoint cost resource, which allows an ALTO server to provide other ALTO resources associated to the endpoint cost resource in the same response.

7.2.1. Media Type

The media type of the multipart endpoint cost resource is "multipart/related;type=application/alto-endpointcost+json".

7.2.2. HTTP Method

The multipart endpoint cost resource is requested using the HTTP POST method.

7.2.3. Accept Input Parameters

The input parameters of the multipart endpoint cost resource are supplied in the body of an HTTP POST request. This document extends the input parameters to an endpoint cost map with a data format indicated by the media type "application/alto-endpointcostparams+json", which is a JSON object of type PVEndpointCostParams, where

```
object {
  [EntityPropertyName ane-property-names<0..*>;]
} PVReqEndpointcost : ReqEndpointcost;
```

with fields:

ane-property-names: This document defines the "ane-property-names" in PVReqEndpointcost as the same as in PVReqFilteredCostMap. See Section 7.1.3.

Example: Consider the network in Figure 1. If an ALTO client wants to query the "max-reservable-bandwidth" between eh1 and eh2, it can submit the following request.

```
POST /ecs/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;type=application/alto-endpointcost+json,
        application/alto-error+json
Content-Length: [TBD]
Content-Type: application/alto-endpointcostparams+json

{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [ "ipv4:1.2.3.4" ],
    "dsts": [ "ipv4:2.3.4.5" ]
  },
  "ane-property-names": [ "max-reservable-bandwidth" ]
}
```

7.2.4. Capabilities

The capabilities of the multipart endpoint cost resource are defined by a JSON object of type PVEndpointcostCapabilities, which is defined as the same as PVFilteredCostMapCapabilities. See Section 7.1.4.

7.2.5. Uses

If this resource supports "persistent-entity-id", it MUST also include the defining resources of persistent ANEs that may appear in the response.

7.2.6. Response

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

The "Content-Type" header of the response MUST be "multipart/related" as defined by [RFC7285] with the following parameters:

type: The type parameter MUST be "application/alto-endpointcost+json".

start: The start parameter is as defined in Section 7.1.6.

boundary: The boundary parameter is as defined in [RFC2387].

The body consists of two parts:

* The Path Vector part MUST include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" MUST has the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-endpointcost+json".

  The body of the Path Vector part MUST be a JSON object with the same format as defined in Section 11.5.1.6 of [RFC7285]. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned endpoint cost map. The resource ID of the version tag MUST follow the format in Section 5.3.2.

* The Unified Property Map part MUST also include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" MUST has the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-propmap+json".

  The body of the Unified Property Map part MUST be a JSON object with the same format as defined in Section 4.6 of [I-D.ietf-alto-unified-props-new]. The JSON object MUST include the "dependent-vtags" field in the "meta" field. The value of the "dependent-vtags" field MUST be an array of VersionTag objects as defined by Section 10.3 of [RFC7285]. The "vtag" of the Path Vector part MUST be included in the "dependent-vtags". If "persistent-entity-id" is requested, the version tags of the dependent resources that may expose the entities in the response MUST also be included. The PropertyMapData has one member for each ANEName that appears in the Path Vector part, which is an entity identifier belonging to the self-defined entity domain as defined in Section 5.1.2.3 of [I-D.ietf-alto-unified-props-new]. The EntityProps has one member for each property requested by the ALTO client if applicable.

If the "start" parameter is not present, the Path Vector part MUST be the first part in the multipart response.

Example: Consider the network in Figure 1. The response of the example request in Section 7.2.3 is as follows.

```
HTTP/1.1 200 OK
Content-Length: [TBD]
Content-Type: multipart/related; boundary=example-1;
              type=application/alto-endpointcost+json

--example-1
Resource-Id: ecs
Content-Type: application/alto-endpointcost+json

{
  "meta": {
    "vtag": {
      "resource-id": "ecs-pv.costmap",
      "tag": "d827f484cb66ce6df6b5077cb8562b0a"
    },
    "dependent-vtags": [
      {
        "resource-id": "my-default-networkmap",
        "tag": "75ed013b3cb58f896e839582504f6228"
      }
    ],
    "cost-type": { "cost-mode": "array", "cost-metric": "ane-path" }
  },
  "cost-map": {
    "ipv4:1.2.3.4": { "ipv4:2.3.4.5": ["ANE1"] }
  }
}
--example-1
Resource-Id: propmap
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "ecs-pv.costmap",
        "tag": "d827f484cb66ce6df6b5077cb8562b0a"
      }
    ]
  },
  "property-map": {
    ".ane:ANE1": { "max-reservable-bandwidth": 100000000 }
  }
}
```

8. Examples

This section lists some examples of Path Vector queries and the corresponding responses. Some long lines are truncated for better readability.

8.1. Example: Information Resource Directory

To give a comprehensive example of the Path Vector extension, we consider the network in Figure 5. The example ALTO server provides the following information resources:

* "my-default-networkmap": A Network Map resource which contains the PIDs in the network.

* "filtered-cost-map-pv": A Multipart Filtered Cost Map resource for Path Vector, which exposes the "max-reservable-bandwidth" property for the PIDs in "my-default-networkmap".

* "ane-props": A filtered Unified Property resource that exposes the information for persistent ANEs in the network.

* "endpoint-cost-pv": A Multipart Endpoint Cost Service for Path Vector, which exposes the "max-reservable-bandwidth" and the "persistent-entity-id" properties.

* "update-pv": An Update Stream service, which provides the incremental update service for the "endpoint-cost-pv" service.

Below is the Information Resource Directory of the example ALTO server. To enable the Path Vector extension, the "path-vector" cost type (Section 6.5) is defined in the "cost-types" of the "meta" field, and is included in the "cost-type-names" of resources "filetered-cost-map-pv" and "endpoint-cost-pv".


```
{
  "meta": {
    "cost-types": {
      "path-vector": {
        "cost-mode": "array",
        "cost-metric": "ane-path"
      }
    }
  },
  "resources": {
    "my-default-networkmap": {
      "uri" : "https://alto.example.com/networkmap",
      "media-type" : "application/alto-networkmap+json"
```

```
        },
        "filtered-cost-map-pv": {
          "uri": "https://alto.example.com/costmap/pv",
          "media-type": "multipart/related;
                         type=application/alto-costmap+json",
          "accepts": "application/alto-costmapfilter+json",
          "capabilities": {
            "cost-type-names": [ "path-vector" ],
            "ane-property-names": [ "max-reservable-bandwidth" ]
          },
          "uses": [ "my-default-networkmap" ]
        },
        "ane-props": {
          "uri": "https://alto.example.com/ane-props",
          "media-type": "application/alto-propmap+json",
          "accepts": "application/alto-propmapparams+json",
          "capabilities": {
            "mappings": {
              ".ane": [ "cpu" ]
            }
          }
        },
        "endpoint-cost-pv": {
          "uri": "https://alto.exmaple.com/endpointcost/pv",
          "media-type": "multipart/related;
                         type=application/alto-endpointcost+json",
          "accepts": "application/alto-endpointcostparams+json",
          "capabilities": {
            "cost-type-names": [ "path-vector" ],
            "ane-property-names": [
              "max-reservable-bandwidth", "persistent-entity-id"
            ]
          },
          "uses": [ "ane-props" ]
        },
        "update-pv": {
          "uri": "https://alto.example.com/updates/pv",
          "media-type": "text/event-stream",
          "uses": [ "endpoint-cost-pv" ],
          "accepts": "application/alto-updatestreamparams+json",
          "capabilities": {
            "support-stream-control": true
          }
        }
      }
    }
```

8.2. Example: Multipart Filtered Cost Map

The following examples demonstrate the request to the "filtered-cost-map-pv" resource and the corresponding response.

The request uses the "path-vector" cost type in the "cost-type" field. The "ane-property-names" field is missing, indicating that the client only requests for the Path Vector but not the ANE properties.

The response consists of two parts. The first part returns the array of ANEName for each source and destination pair. There are two ANEs, where "L1" represents the interconnection link L1, and "L2" represents the interconnection link L2.

The second part returns an empty Property Map. Note that the ANE entries are omitted since they have no properties (See Section 3.1 of [I-D.ietf-alto-unified-props-new]).

```
POST /costmap/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;type=application/alto-costmap+json,
        application/alto-error+json
Content-Length: [TBD]
Content-Type: application/alto-costmapfilter+json

{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "pids": {
    "srcs": [ "PID1" ],
    "dsts": [ "PID3", "PID4" ]
  }
}

HTTP/1.1 200 OK
Content-Length: [TBD]
Content-Type: multipart/related; boundary=boundary;
              type=application/alto-costmap+json

--boundary
Resource-Id: costmap
Content-Type: application/alto-costmap+json

{
  "meta": {
```

```
        "vtag": {
          "resource-id": "filtered-cost-map-pv.costmap",
          "tag": "d827f484cb66ce6df6b5077cb8562b0a"
        },
        "dependent-vtags": [
          {
            "resource-id": "my-default-networkmap",
            "tag": "75ed013b3cb58f896e839582504f6228"
          }
        ],
        "cost-type": {
          "cost-mode": "array",
          "cost-metric": "ane-path"
        }
      },
      "cost-map": {
        "PID1": {
          "PID3": [ "L1" ],
          "PID4": [ "L1", "L2" ]
        }
      }
    }
    --boundary
    Resource-Id: propmap
    Content-Type: application/alto-propmap+json

    {
      "meta": {
        "dependent-vtags": [
          {
            "resource-id": "filtered-cost-map-pv.costmap",
            "tag": "d827f484cb66ce6df6b5077cb8562b0a"
          }
        ]
      },
      "property-map": {
      }
    }
```

8.3. Example: Multipart Endpoint Cost Resource

The following examples demonstrate the request to the "endpoint-cost-pv" resource and the corresponding response.

The request uses the path vector cost type in the "cost-type" field, and queries the Maximum Reservable Bandwidth ANE property and the Persistent Entity property.

The response consists of two parts. The first part returns the array of ANEName for each valid source and destination pair, where "NET1" represent sub-network NET1, and "AGGR" is the aggregation of L1 and NET3.

The second part returns the requested properties of ANEs. Since NET1 has sufficient bandwidth, it sets the "max-reservable-bandwidth" to a sufficiently large number. It also represents a persistent ANE defined in the "ane-props" resource, identified by "ane-props.ane:datacenter1". The aggregated "max-reservable-bandwidth" of ane:AGGR is constrained by the link capacity of L1. The "persistent-entity-id" property is omitted as both L1 and NET3 do not represent any persistent entity.

```
POST /endpointcost/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;
        type=application/alto-endpointcost+json,
        application/alto-error+json
Content-Length: [TBD]
Content-Type: application/alto-endpointcostparams+json

{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [ "ipv4:1.2.3.4", "ipv4:2.3.4.5" ],
    "dsts": [ "ipv4:3.4.5.6" ]
  },
  "ane-property-names": [
    "max-reservable-bandwidth",
    "persistent-entity-id"
  ]
}

HTTP/1.1 200 OK
Content-Length: [TBD]
Content-Type: multipart/related; boundary=boundary;
              type=application/alto-endpointcost+json

--boundary
Resource-Id: ecs
Content-Type: application/alto-endpointcost+json

{
  "meta": {
```

```
        "vtags": {
          "resource-id": "endpoint-cost-pv.ecs",
          "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
        },
        "cost-type": {
          "cost-mode": "array",
          "cost-metric": "ane-path"
        }
      },
      "endpoint-cost-map": {
        "ipv4:1.2.3.4": {
          "ipv4:3.4.5.6":   [ "NET1", "AGGR" ]
        },
        "ipv4:2.3.4.5": {
          "ipv4:3.4.5.6":   [ "NET1", "AGGR" ]
        }
      }
    }
    --boundary
    Resource-Id: propmap
    Content-Type: application/alto-propmap+json

    {
      "meta": {
        "dependent-vtags": [
          {
            "resource-id": "endpoint-cost-pv.ecs",
            "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
          },
          {
            "resource-id": "ane-props",
            "tag": "bf3c8c1819d2421c9a95a9d02af557a3"
          }
        ]
      },
      "property-map": {
        ".ane:NET1": {
          "max-reservable-bandwidth": 50000000000,
          "persistent-entity-id": "ane-props.ane:datacenter1",
        },
        ".ane:AGGR": {
          "max-reservable-bandwidth": 10000000000
        }
      }
    }
```


After the client obtains "ane-props.ane:datacenter1", it can query the "ane-props" resource to get the properties of the persistent ANE.

8.4. Example: Incremental Updates

In this example, an ALTO client subscribes to the incremental update for the multipart endpoint cost resource "endpoint-cost-pv".

```
POST /updates/pv HTTP/1.1
Host: alto.example.com
Accept: text/event-stream
Content-Type: application/alto-updatestreamparams+json
Content-Length: [TBD]

{
  "add": {
    "ecspvsub1": {
      "resource-id": "endpoint-cost-pv",
      "input": <ecs-input>
    }
  }
}
```

Based on the server-side process defined in [I-D.ietf-alto-incr-update-sse], the ALTO server will send the "control-uri" first using Server-Sent Event (SSE), followed by the full response of the multipart message.

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

event: application/alto-updatestreamcontrol+json
data: {"control-uri": "https://alto.example.com/updates/streams/123"}

event: multipart/related;boundary=boundary;
       type=application/alto-endpointcost+json,ecspvsub1
data: --boundary
data: Resource-ID: ecsmap
data: Content-Type: application/alto-endpointcost+json
data:
data: <endpoint-cost-map-entry>
data: --boundary
data: Resource-ID: propmap
data: Content-Type: application/alto-propmap+json
data:
data: <property-map-entry>
data: --boundary--
```

When the contents change, the ALTO server will publish the updates for each node in this tree separately.

```
event: application/merge-patch+json, ecspvsub1.ecsmap
data: <Merge patch for endpoint-cost-map-update>

event: application/merge-patch+json, ecspvsub1.propmap
data: <Merge patch for property-map-update>
```

9. Compatibility

9.1. Compatibility with Legacy ALTO Clients/Servers

The multipart filtered cost map resource and the multipart endpoint cost resource has no backward compatibility issue with legacy ALTO clients and servers. Although these two types of resources reuse the media types defined in the base ALTO protocol for the accept input parameters, they have different media types for responses. If the ALTO server provides these two types of resources, but the ALTO client does not support them, the ALTO client will ignore the resources without conducting any incompatibility.

9.2. Compatibility with Multi-Cost Extension

This document does not specify how to integrate the Path Vector cost type with the multi-cost extension [RFC8189]. While it is not RECOMMENDED to put the Path Vector cost type with other cost types in a single query, there is no compatible issue.

9.3. Compatibility with Incremental Update

The extension specified in this document is not compatible with the original incremental update extension [I-D.ietf-alto-incr-update-sse]. A legacy ALTO client cannot recognize the compound client-id, and a legacy ALTO server may use the same client-id for updates of both parts.

ALTO clients and servers must follow the specifications given in this document to ensure compatibility with the incremental update extension.

9.4. Compatibility with Cost Calendar

The extension specified in this document is compatible with the Cost Calendar extension [I-D.ietf-alto-cost-calendar]. When used together with the Cost Calendar extension, the cost value between a source and a destination is an array of path vectors, where the k-th path vector refers to the abstract network paths traversed in the k-th time interval by traffic from the source to the destination.

When used with time-varying properties, e.g., maximum reservable bandwidth (maxresbw), a property of a single entity may also have different values in different time intervals. In this case, an ANE with different property values must be considered as different ANEs.

The two extensions combined together can provide the historical network correlation information for a set of source and destination pairs. A network broker or client may use this information to derive other resource requirements such as Time-Block-Maximum Bandwidth, Bandwidth-Sliding-Window, and Time-Bandwidth-Product (TBP) (See [SENSE] for details).

10. General Discussions

10.1. Constraint Tests for General Cost Types

The constraint test is a simple approach to query the data. It allows users to filter the query result by specifying some boolean tests. This approach is already used in the ALTO protocol. [RFC7285] and [RFC8189] allow ALTO clients to specify the "constraints" and "or-constraints" tests to better filter the result.

However, the current syntax can only be used to test scalar cost types, and cannot easily express constraints on complex cost types, e.g., the Path Vector cost type defined in this document.

In practice, developing a language for general-purpose boolean tests can be complex and is likely to be a duplicated work. Thus, it is worth looking into the direction of integrating existing well-developed query languages, e.g., XQuery and JSONiq, or their subset with ALTO.

Filtering the Path Vector results or developing a more sophisticated filtering mechanism is beyond the scope of this document.

10.2. General Multipart Resources Query

Querying multiple ALTO information resources continuously may be a general requirement. And the coming issues like inefficiency and inconsistency are also general. There is no standard solving these issues yet. So we need some approach to make the ALTO client request the compound ALTO information resources in a single query.

11. Security Considerations

This document is an extension of the base ALTO protocol, so the Security Considerations [RFC7285] of the base ALTO protocol fully apply when this extension is provided by an ALTO server.

The Path Vector extension requires additional considerations on two security considerations discussed in the base protocol: confidentiality of ALTO information (Section 15.3 of [RFC7285]) and availability of ALTO service (Section 15.5 of [RFC7285]).

For confidentiality of ALTO information, a network operator should be aware of that this extension may introduce a new risk: the Path Vector information may make network attacks easier. For example, as the Path Vector information may reveal more fine-grained internal network structures than the base protocol, an ALTO client may detect the bottleneck link and start a distributed denial-of-service (DDoS) attack involving minimal flows to conduct the in-network congestion.

To mitigate this risk, the ALTO server should consider protection mechanisms to reduce information exposure or obfuscate the real information, in particular, in settings where the network and the application do not belong to the same trust domain. But the implementation of Path Vector extension involving reduction or obfuscation should guarantee the requested properties are still accurate.

For availability of ALTO service, an ALTO server should be cognizant that using Path Vector extension might have a new risk: frequent requesting for path vectors might conduct intolerable increment of the server-side storage and break the ALTO server. It is known that the computation of Path Vectors is unlikely to be cacheable, in that the results will depend on the particular requests (e.g., where the flows are distributed). Hence, the service providing Path Vectors may become an entry point for denial-of-service attacks on the availability of an ALTO server. To avoid this risk, authenticity and authorization of this ALTO service may need to be better protected.

12. IANA Considerations

12.1. ALTO Entity Domain Registry

This document registers a new entry to the ALTO Domain Entity Registry, as instructed by Section 12.2 of [I-D.ietf-alto-unified-props-new]. The new entry is as shown below in Table 1.

Identifier	Entity Address Encoding	Hierarchy & Inheritance	Media Type of Defining Resource
ane	See Section 6.2.2	None	application/alto-propmap+json

Table 1: ALTO Entity Domain

Identifier: See Section 6.2.1.

Entity Identifier Encoding: See Section 6.2.2.

Hierarchy: None

Inheritance: None

Media Type of Defining Resource: See Section 6.2.4.

Security Considerations: In some usage scenarios, ANE addresses carried in ALTO Protocol messages may reveal information about an ALTO client or an ALTO service provider. Applications and ALTO service providers using addresses of ANEs will be made aware of how (or if) the addressing scheme relates to private information and network proximity, in further iterations of this document.

12.2. ALTO Entity Property Type Registry

Two initial entries are registered to the ALTO Domain "ane" in the "ALTO Entity Property Type Registry", as instructed by Section 12.3 of [I-D.ietf-alto-unified-props-new]. The two new entries are shown below in Table 2.

Identifier	Intended Semantics
max-reservable-bandwidth	See Section 6.4.1
persistent-entity-id	See Section 6.4.2

Table 2: Initial Entries for ane Domain in the ALTO Entity Property Types Registry

13. Acknowledgments

The authors would like to thank discussions with Andreas Voellmy, Erran Li, Haibin Song, Haizhou Du, Jiayuan Hu, Qiao Xiang, Tianyuan Liu, Xiao Shi, Xin Wang, and Yan Luo. The authors thank Greg Bernstein (Grotto Networks), Dawn Chen (Tongji University), Wendy Roome, and Michael Scharf for their contributions to earlier drafts.

14. References

14.1. Normative References

[I-D.ietf-alto-cost-calendar]
Randriamasy, S., Yang, Y., WU, Q., Lingli, D., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", Work in Progress, Internet-Draft, draft-ietf-alto-cost-calendar-21, 17 March 2020, <http://www.ietf.org/internet-drafts/draft-ietf-alto-cost-calendar-21.txt>.

[I-D.ietf-alto-incr-update-sse]
Roome, W. and Y. Yang, "ALTO Incremental Updates Using Server-Sent Events (SSE)", Work in Progress, Internet-Draft, draft-ietf-alto-incr-update-sse-22, 20 March 2020, <http://www.ietf.org/internet-drafts/draft-ietf-alto-incr-update-sse-22.txt>.

[I-D.ietf-alto-performance-metrics]
WU, Q., Yang, Y., Lee, Y., Dhody, D., Randriamasy, S., and L. Contreras, "ALTO Performance Cost Metrics", Work in Progress, Internet-Draft, draft-ietf-alto-performance-metrics-11, 12 June 2020, <http://www.ietf.org/internet-drafts/draft-ietf-alto-performance-metrics-11.txt>.

[I-D.ietf-alto-unified-props-new]
Roome, W., Randriamasy, S., Yang, Y., Zhang, J., and K. Gao, "Unified Properties for the ALTO Protocol", Work in Progress, Internet-Draft, draft-ietf-alto-unified-props-new-11, 9 March 2020, <http://www.ietf.org/internet-drafts/draft-ietf-alto-unified-props-new-11.txt>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, DOI 10.17487/RFC2387, August 1998, <https://www.rfc-editor.org/info/rfc2387>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <https://www.rfc-editor.org/info/rfc8189>.

14.2. Informative References

[AAAI2019] Xiang, Q., Yu, H., Aspnes, J., Le, F., Kong, L., and Y.R. Yang, "Optimizing in the dark: Learning an optimal solution through a simple request interface", Proceedings of the AAAI Conference on Artificial Intelligence 33, 1674-1681 , 2019.

[I-D.bernstein-alto-topo]
Bernstein, G., Yang, Y., and Y. Lee, "ALTO Topology Service: Uses Cases, Requirements, and Framework", Work in Progress, Internet-Draft, draft-bernstein-alto-topo-00, 21 October 2013, <http://www.ietf.org/internet-drafts/draft-bernstein-alto-topo-00.txt>.

[I-D.contreras-alto-service-edge]
Contreras, L., Perez, D., and C. Rothenberg, "Use of ALTO for Determining Service Edge", Work in Progress, Internet-Draft, draft-contreras-alto-service-edge-00, 4 November 2019, <http://www.ietf.org/internet-drafts/draft-contreras-alto-service-edge-00.txt>.

[I-D.huang-alto-mowie-for-network-aware-app]
Huang, W., Zhang, Y., Yang, R., Xiong, C., Lei, Y., Han, Y., and G. Li, "MoWIE for Network Aware Application", Work in Progress, Internet-Draft, draft-huang-alto-mowie-for-network-aware-app-00, 9 March 2020, <http://www.ietf.org/internet-drafts/draft-huang-alto-mowie-for-network-aware-app-00.txt>.

[I-D.ietf-dmm-5g-uplane-analysis]
Homma, S., Miyasaka, T., Matsushima, S., and D. Voyer, "User Plane Protocol and Architectural Analysis on 3GPP 5G System", Work in Progress, Internet-Draft, draft-ietf-dmm-5g-uplane-analysis-03, 3 November 2019, <http://www.ietf.org/internet-drafts/draft-ietf-dmm-5g-uplane-analysis-03.txt>.

[I-D.yang-alto-deliver-functions-over-networks]
Yang, S., Cui, L., Xu, M., Yang, Y., and R. Huang, "Delivering Functions over Networks: Traffic and Performance Optimization for Edge Computing using ALTO", Work in Progress, Internet-Draft, draft-yang-alto-deliver-functions-over-networks-01, 13 July 2020, <http://www.ietf.org/internet-drafts/draft-yang-alto-deliver-functions-over-networks-01.txt>.

[LHC] "CERN - LHC", 2019, <https://atlas.cern/tags/lhc>.

[SENSE] "Services - SENSE", 2019, <http://sense.es.net/services>.

[TON2019] Gao, K., Xiang, Q., Wang, X., Yang, Y.R., and J. Bi, "An objective-driven on-demand network abstraction for adaptive applications", IEEE/ACM Transactions on Networking (TON) Vol 27, no. 2 (2019): 805-818., 2019.

Appendix A. Changes since -10

Revision -11

* replaces "part" with "components" in the abstract;

* identifies additional requirements (AR) derived from the flow scheduling example, and introduces how the extension addresses the additional requirements

* fixes the inconsistent use of "start" parameter in multipart responses;

* specifies explicitly how to handle "cost-constraints";

* uses the latest IANA registration mechanism defined in [I-D.ietf-alto-unified-props-new];

* renames "persistent-entities" to "persistent-entity-id";

* makes "application/alto-propmap+json" as the media type of defining resources for the "ane" domain;

* updates the examples;

* adds the discussion on ephemeral and persistent ANEs.

Appendix B. Changes since -09

Revision -10

* revises the introduction which

  - extends the scope where the PV extension can be applied beyond the "path correlation" information

* brings back the capacity region use case to better illustrate the problem

* revises the overview to explain and defend the concepts and decision choices

* fixes inconsistent terms, typos

Appendix C. Changes since -08

This revision

* fixes a few spelling errors

* emphasizes that abstract network elements can be generated on demand in both introduction and motivating use cases

Appendix D. Changes Since Version -06

* We emphasize the importance of the path vector extension in two aspects:

  1. It expands the problem space that can be solved by ALTO, from preferences of network paths to correlations of network paths.

  2. It is motivated by new usage scenarios from both application's and network's perspectives.

* More use cases are included, in addition to the original capacity region use case.

* We add more discussions to fully explore the design space of the path vector extension and justify our design decisions, including the concept of abstract network element, cost type (reverted to -05), newer capabilities and the multipart message.

* Fix the incremental update process to be compatible with SSE -16 draft, which uses client-id instead of resource-id to demultiplex updates.

* Register an additional ANE property (i.e., persistent-entities) to cover all use cases mentioned in the draft.

Authors' Addresses

Kai Gao
China
610000
Chengdu
No.24 South Section 1, Yihuan Road
Sichuan University

Email: kaigao@scu.edu.cn

Young Lee

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
91460 Nozay
France

Email: sabine.randriamasy@nokia-bell-labs.com

Yang Richard Yang
Yale University
51 Prospect Street
New Haven, CT
United States of America

Email: yry@cs.yale.edu

Jingxuan Jensen Zhang
China
201804
Shanghai
4800 Caoan Road
Tongji University

Email: jingxuan.n.zhang@gmail.com

ALTO Working Group Q. Wu
Internet-Draft Huawei
Intended status: Standards Track Y. Yang
Expires: January 13, 2021 Yale University
Y. Lee
Samsung
D. Dhody
Huawei
S. Randriamasy
Nokia Bell Labs
L. Contreras
Telefonica
July 12, 2020

ALTO Performance Cost Metrics
draft-ietf-alto-performance-metrics-12

Abstract

Cost metric is a basic concept in Application-Layer Traffic Optimization (ALTO), and is used in basic ALTO services including both the cost map service and the endpoint cost service.

Different applications may use different cost metrics, but the ALTO base protocol [RFC7285] defines only a single cost metric, i.e., the generic "routingcost" metric; see Sec. 14.2 of [RFC7285]. Hence, if the ALTO client of an application wants to issue a cost map or an endpoint cost request to determine the resource provider that offers better delay performance (i.e., low-delay) to a resource consumer, the base protocol does not define the cost metric to be used.

This document addresses the issue by introducing network performance metrics, including network delay, jitter, packet loss rate, hop count, and bandwidth. The ALTO server may derive and aggregate such performance metrics from routing protocols such as BGP-LS, OSPF-TE and ISIS-TE, or from end-to-end traffic management tools, and then expose the information to allow applications to determine "where" to connect based on network performance criteria.

There are multiple sources to derive the performance metrics. For example, whether the metric reported is an estimation based on measurements or it is a service-level agreement (SLA) can define the meaning of the performance metric. Hence, an application may need additional contextual information beyond the metric value. This document introduces an additional "cost-context" field to the ALTO "cost-type" field to convey such information.

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 [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 13, 2021.

Copyright Notice

Table of Contents

1. Introduction

Cost Metric is a basic concept in Application-Layer Traffic Optimization (ALTO). It is used in both the ALTO cost map service and the ALTO endpoint cost service in the ALTO base protocol [RFC7285].

Since different applications may use different cost metrics, the ALTO base protocol introduces an ALTO Cost Metric Registry (Section 14.2 of [RFC7285]), as a systematic mechanism to allow different metrics to be specified. For example, a delay-sensitive application may want to use latency related metrics, and a bandwidth-sensitive application may want to use bandwidth related metrics. The ALTO base protocol, however, has registered only a single cost metric, i.e., the generic "routingcost" metric; no latency or bandwidth related metrics are defined.

This document registers a set of new cost metrics specified in Table 1, to allow applications to better determine "where" to connect based on network performance criteria. This document follows the guideline defined in Section 14.2 of the ALTO base protocol [RFC7285]) on registering ALTO cost metrics. Hence it specifies the identifier, the intended semantics, and the security considerations of each one of the metrics defined in Table 1.

Metric	Definition	Origin Example
One-way Delay	Section 3.1	[RFC7679]
Round-trip Delay	Section 3.2	[RFC2681]
Delay Variation	Section 3.3	[RFC3393]
Hop Count	Section 3.4	[RFC7285]
Loss Rate	Section 3.5	[RFC7680]
TCP Throughput	Section 4.1	[RFC6349]
Residue Bandwidth	Section 4.2	[RFC7810]
Max Reservable Bandwidth	Section 4.3	[RFC5305]

Table 1. Cost Metrics Defined in this Document.

The purpose of this document is to ensure proper usage of the performance metrics defined in Table 1; it does not claim novelty of the metrics. The Origin column of Table 1 gives the RFC which defines each metric.

We can rough classify the performance metrics into two categories: those derived from the performance of individual packets (i.e., one-way delay, round-trip delay, delay variation, hop count, and loss rate), and those related with bandwidth (TCP throughput, residue bandwidth and max reservable bandwidth). These two categories are defined in Section 3 and Section 4 respectively. Note that all metrics except round trip delay are unidirectional. Hence, a client will need to query both directions if needed.

An ALTO server may provide only a subset of the metrics described in this document. For example, those that are subject to privacy concerns should not be provided to unauthorized ALTO clients. Hence, all cost metrics defined in this document are optional and not all of them need to be exposed to a given application. When an ALTO server supports a cost metric defined in this document, it should announce this metric in its information resource directory (IRD).

An ALTO server introducing these metrics should consider security issues. As a generic security consideration on the reliability and trust in the exposed metric values, applications SHOULD rapidly give up using ALTO-based guidance if they detect that the exposed information does not preserve their performance level or even degrades it. This document discusses security considerations in more details in Section 6.

Following the ALTO base protocol, this document uses JSON to specify the value type of each defined metric. See [RFC8259] for JSON data type specification.

2. Performance Metric Attributes

When defining the metrics in Table 1, this document considers the guidelines specified in [RFC6390], which requires fine-grained specification of (i) Metric Name, (ii) Metric Description, (iii) Method of Measurement or Calculation, (iv) Units of Measurement, (v) Measurement Points, and (vi) Measurement Timing. In particular, for each metric, this document defines (i) Metric Name, (ii) Metric Description, and (iv) Units of Measurement. The Measurement Points are always specified by the specific ALTO services; for example, endpoint cost service is between the two end points.

On the other hand, to be able to use coarse-grained information such as routing system information (e.g., [RFC8571]), which may not provide fine-grained information such as (iii) Method of Measurement or Calculation and (vi) Measurement Timing, this document provides context information to indicate the source of information and hence available metric details.

2.1. Performance Metric Context: cost-context

The details of a performance metric depend on the source of the information. Specifically, this document defines four types of information sources: "nominal", and "sla" (service level agreement), "import", and "estimation".

For a given type of source, precise interpretation of a performance metric value can depend on particular measurement and computation parameters. For example, see Section 3.8 of [RFC7679] on items which a more complete measurement-based report should include.

To make it possible to specify the source and the aforementioned parameters, this document introduces an optional "cost-context" field to the "cost-type" field defined by the ALTO base protocol (Section 10.7 of [RFC7285]) as the following:

```
object {
  CostMetric   cost-metric;
  CostMode     cost-mode;
  [CostContext cost-context;]
  [JSONString  description;]
} CostType;

object {
  JSONString    cost-source;
  [JSONValue    parameters;]
} CostContext;
```

The "cost-source" field of the "cost-context" field MUST be one of four category values: "nominal", "sla", "import", and "estimation". "cost-context" will not be used as a key to distinguish among performance metrics. Hence, an ALTO information resource SHOULD NOT announce multiple CostType with the same "cost-metric" and "cost-mode". They can be placed into different information resources.

The "nominal" category indicates that the value of the metric is statically configured by the underlying devices. Not all metrics have reasonable "nominal" values. For example, throughput can have a nominal value, which indicates the configured transmission rate of the devices; latency typically do not have a nominal value.

The "sla" category indicates that the value of the metric is derived from some commitment which this document refers to as service-level agreement (SLA). Some operators also use terms such as "target" or "committed" values. For a "sla" metric, it is RECOMMENDED that the "parameters" field provides a link to the SLA definition.

The "estimation" category indicates that the value of the metric is computed through an estimation process. An ALTO server may compute "estimation" values by retrieving and/or aggregating information from routing protocols (e.g., [RFC8571]) and traffic measurement management tools (e.g., TWAMP [RFC5357]), with corresponding operational issues. A potential architecture on estimating these metrics is shown in Figure 1 below. Section 5 will discuss in more detail the operational issues and how a network may address them.

```
 +--------+   +--------+  +--------+
 | Client |   | Client |  | Client |
 +----^---+   +---^----+  +---^----+
      |           |           |
      +-----------|-----------+
            NBI   |ALTO protocol
                  |
                  |
               +--+-----+  retrieval      +-----------+
               |  ALTO  |<----------------| Routing   |
               | Server |  and aggregation|           |
               |        |<-------------+  | Protocols |
               +--------+              |  +----------+
                                       |
                                       |  +-----------+
                                       |  |Management |
                                       ---|           |
                                          |  Tool     |
                                          +-----------+
```

Figure 1. Potential framework to compute estimation to performance metrics

A particular type of "estimation is direct "import", which indicates that the value of the metric is imported directly from a specific existing protocol or system. Specifying "import" as source instead of the more generic "estimation" may allow better tracing of information flow. For an "import" metric, it is RECOMMENDED that the "parameters" field provides details to the system from which raw data is imported. In particular, one may notice that the set of end-to-end metrics defined in Table 1 has large overlap with the set defined in [RFC8571], in the setting of IGP traffic engineering performance metrics for each link (i.e., unidirectional link delay, min/max unidirectional link delay, unidirectional delay variation, unidirectional link loss, unidirectional residual bandwidth, unidirectional available bandwidth, unidirectional utilized bandwidth). Hence, an ALTO server may use "import" to indicate that its end-to-end metrics are computed from link metrics imported from [RFC8571].

There can be overlap in deciding the cost-source category. It is the operator of an ALTO server who chooses the category. If a metric does not include a "cost-source" value, the application MUST assume that the value of "cost-source" is the most generic "estimation".

2.2. Performance Metric Statistics

The measurement of a performance metric often yields a set of samples from an observation distribution ([Prometheus], instead of a single value. This document considers that the samples are aggregated as a statistic when reported. Hence, each performance metric's identifier should indicate the statistic (i.e., an aggregation operation), to become <metric-base-identifier>-<stat>, where <stat> MUST be one of the following:

percentile, with letter p followed by a number p:

> gives the p percentile. Specifically, consider the samples coming from a random variable X. The metric returns x, relative to 100, such that the probability of X is less than or equal to x, i.e., Prob(X <= x) = p/100. The number p MUST be a non-negative JSON number in the range [0, 100] (i.e., greater than or equal to 0 and less than or equal to 100). To avoid complex identifiers, the number MUST NOT include the minus or the exp component (Section 6 of [RFC8259]). For example, delay-ow-p75 gives the 75% percentile of observed one-way delay; delay-ow-p99.9 gives the 99.9% percentile of delay. Note that some systems use quantile, which is in the range [0, 1]. This document uses percentile to make the identifier easier to read.

min:

> the minimal value of the observation distribution.

max:

> the maximal value of the observations.

median:

the mid point (i.e., p50) of the observation distribution.

mean:

the arithmetic mean value of the observations.

stddev:

the standard deviation of the observations.

stdvar:

the standard variance of the observations.

If a metric has no <stat> (and hence no - as well), the metric MUST be considered as the 50 percentile (median). Since this scheme is common for all metrics defined in this document, below we only specify the base identifier.

3. Packet Performance Metrics

This section introduces ALTO network performance metrics including one way delay, round trip delay, delay variation, hop count, and packet loss rate. They measure the "quality of experience" of the stream of packets sent from a resource provider to a resource consumer. The measures of each individual packet (pkt) can include the delay from the time that the packet enters the network to the time that the packet leaves the network (pkt.delay); the number of network hops that the packet traverses (pkt.hopcount); and whether the packet is dropped before reaching destination (pkt.dropped). The semantics of the performance metrics defined in this section is that they are statistics (percentiles) computed from these measures; for example, the x-percentile of the one-way delay is the x-percentile of the set of delays {pkt.delay} for the packets in the stream.

3.1. Cost Metric: One-Way Delay (delay-ow)

3.1.1. Base Identifier

The base identifier for this performance metric is "delay-ow".

3.1.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of [RFC8259] Section 6. The unit is expressed in milliseconds. Hence, the number can be a floating point number to express delay that is smaller than milliseconds. The number MUST be non-negative.

3.1.3. Intended Semantics and Use

Intended Semantics: To specify spatial and temporal aggregated delay of a stream of packets from the specified source and the specified destination. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 1: Delay value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```


```
{
  "cost-type": {"cost-mode" : "numerical",
                "cost-metric" : "delay-ow"},
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv6:2000::1:2345:6789:abcd"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta" :{
    "cost-type": {"cost-mode" : "numerical",
                  "cost-metric" : "delay-ow"
     }
   },
    "endpoint-cost-map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89"    : 10,
        "ipv4:198.51.100.34" : 20,
        "ipv6:2000::1:2345:6789:abcd"  : 30,
    }
  }
}
```

Comment: Since the "cost-type" does not include the "cost-source" field, the values are based on "estimation". Since the identifier does not include the -<percentile> component, the values will represent median values.

3.1.4. Cost-Context Specification Considerations

"nominal": Typically network one-way delay does not have a nominal value.

"sla": Many networks provide delay in their application-level service level agreements. It is RECOMMENDED that the "parameters" field of an "sla" one-way delay metric provides a link ("link") to the SLA definition.

"import": There can be multiple sources to import one-way delay. For example, if the import is from [RFC8571] (by using unidirectional link delay, min/max unidirectional link delay), it is RECOMMENDED that "parameters" provides "protocol" as a field and "RFC8571" as the value. During import, the server should be cognizant of potential issues when computing an end-to-end summary statistics from a link statistics. Another example import source is the IPPM framework. For IPPM, it is recommended that "parameters" provides "protocol" as a field and "ippm" as the value; see Section 4 of [I-D.ietf-ippm-initial-registry] for additional fields which can be specified for "ippm" in "parameters".

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" one-way delay metric provides a link ("link") to a description of the "estimation" method.

3.2. Cost Metric: Round-trip Delay (delay-rt)

3.2.1. Base Identifier

The base identifier for this performance metric is "delay-rt".

3.2.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of [RFC8259] Section 6. The number MUST be non-negative. The unit is expressed in milliseconds.

3.2.3. Intended Semantics and Use

Intended Semantics: To specify spatial and temporal aggregated round-trip delay between the specified source and specified destination. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Note that it is possible for a client to query two one-way delays and then compute the round-trip delay. The server should be cognizant of the consistency of values.

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 2: Round-trip Delay value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```


```
{
 "cost-type": {"cost-mode" : "numerical",
               "cost-metric" : "delay-rt"},
  "endpoints" : {
     "srcs": [ "ipv4:192.0.2.2" ],
     "dsts": [
       "ipv4:192.0.2.89",
       "ipv4:198.51.100.34",
       "ipv6:2000::1:2345:6789:abcd"
     ]
   }
}
```


```
  HTTP/1.1 200 OK
  Content-Length: TBA
  Content-Type: application/alto-endpointcost+json
```


```
  {
    "meta" :{
      "cost-type": {"cost-mode" : "numerical",
                    "cost-metric" : "delay-rt"
       }
     },
      "endpoint-cost-map" : {
        "ipv4:192.0.2.2": {
          "ipv4:192.0.2.89"    : 4,
          "ipv4:198.51.100.34" : 3,
          "ipv6:2000::1:2345:6789:abcd"  : 2,
      }
    }
  }
```


3.2.4. Cost-Context Specification Considerations

"nominal": Typically network round-trip delay does not have a nominal value.

"sla": It is RECOMMENDED that the "parameters" field of an "sla" round-trip delay metric provides a link ("link") to the SLA definition.

"import": There can be multiple sources to import round-trip delay. If the import is from [RFC8571] (by using unidirectional link delay, min/max unidirectional link delay), it is RECOMMENDED that "parameters" provides "protocol" as a field and "RFC8571" as the value; see Section 3.1.4 for discussions on summing up link metrics to obtain end-to-end metrics. If the import is from the IPPM framework, it is recommended that "parameters" provides "protocol" as a field and "ippm" as the value; see Section 4 of [I-D.ietf-ippm-initial-registry] for additional fields which can be specified for "ippm" in "parameters".

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" round-trip delay metric provides a link ("link") to a description of the "estimation" method.

3.3. Cost Metric: Delay Variation (delay-variation)

3.3.1. Base Identifier

The base identifier for this performance metric is "delay-variation".

3.3.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of [RFC8259] Section 6. The number MUST be non-negative. The unit is expressed in milliseconds.

3.3.3. Intended Semantics and Use

Intended Semantics: To specify spatial and temporal aggregated delay variation (also called delay jitter)) with respect to the minimum delay observed on the stream over the specified source and destination. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Note that in statistics, variations are typically evaluated by the distance from samples relative to the mean. In networking context, it is more commonly defined from samples relative to the min. This definition follows the networking convention.

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 3: Delay variation value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
   application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type": {"cost-mode" : "numerical",
   "cost-metric" : "delay-var"},
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv6:2000::1:2345:6789:abcd"
    ]
  }
}
HTTP/1.1 200 OK
 Content-Length: TBA
 Content-Type: application/alto-endpointcost+json
{
  "meta": {
           "cost type": {
           "cost-mode": "numerical",
           "cost-metric":"delay-var"
    }
   },
  "endpoint-cost-map": {
           "ipv4:192.0.2.2": {
           "ipv4:192.0.2.89"    : 0
           "ipv4:198.51.100.34" : 1
           "ipv6:2000::1:2345:6789:abcd"  : 5
         }
      }
   }
```

3.3.4. Cost-Context Specification Considerations

"nominal": Typically network delay variation does not have a nominal value.

"sla": It is RECOMMENDED that the "parameters" field of an "sla" delay variation metric provides a link ("link") to the SLA definition.

"import": There can be multiple sources to import delay variation. If the import is from [RFC8571] (by using unidirectional delay variation), it is RECOMMENDED that "parameters" provides "protocol" as a field and "RFC8571" as the value; see Section 3.1.4 for discussions on summing up link metrics to obtain end-to-end metrics. If the import is from the IPPM framework, it is recommended that "parameters" provides "protocol" as a field and "ippm" as the value; see Section 4 of [I-D.ietf-ippm-initial-registry] for additional fields which can be specified for "ippm" in "parameters".

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" delay variation metric provides a link ("link") to a description of the "estimation" method.

3.4. Cost Metric: Hop Count (hopcount)

The metric hopcount is mentioned in [RFC7285] Section 9.2.3 as an example. This section further clarifies its properties.

3.4.1. Base Identifier

The base identifier for this performance metric is "hopcount".

3.4.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of [RFC8259] Section 6. The number MUST be a non-negative integer (greater than or equal to 0). The value represents the number of hops.

3.4.3. Intended Semantics and Use

Intended Semantics: To specify the number of hops in the path from the specified source to the specified destination. The hop count is a basic measurement of distance in a network and can be exposed as router hops, in direct relation to the routing protocols originating this information. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 4: hopcount value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

  {
    "cost-type": {"cost-mode" : "numerical",
     "cost-metric" : "hopcount"},
    "endpoints" : {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv6:2000::1:2345:6789:abcd"
      ]
    }
  }

HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-endpointcost+json
{
    "meta": {
               "cost type": {
             "cost-mode": "numerical",
             "cost-metric":"hopcount"}
       }
    },
   "endpoint-cost-map": {
           "ipv4:192.0.2.2": {
           "ipv4:192.0.2.89"   : 5,
           "ipv4:198.51.100.34": 3,
           "ipv6:2000::1:2345:6789:abcd" : 2,
                            }
             }
 }
```

3.4.4. Cost-Context Specification Considerations

"nominal": Typically hop count does not have a nominal value.

"sla": Typically hop count does not have an SLA value.

"import": There can be multiple sources to import hop count such as IGP routing protocols.

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" hop count metric provides a link ("link") to a description of the "estimation" method.

3.5. Cost Metric: Loss Rate (lossrate)

3.5.1. Base Identifier

The base identifier for this performance metric is "lossrate".

3.5.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of [RFC8259] Section 6. The number MUST be non-negative. The value represents the percentage of packet losses.

3.5.3. Intended Semantics and Use

Intended Semantics: To specify spatial and temporal aggregated packet loss rate from the specified source and the specified destination. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 5: Loss rate value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

  {
    "cost-type": {"cost-mode" : "numerical",
                  "cost-metric" : "lossrate"
    },
    "endpoints" : {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv6:2000::1:2345:6789:abcd"
      ]
    }
  }

HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-endpointcost+json
{
    "meta": {
      "cost-type": {
        "cost-mode": "numerical",
        "cost-metric":"lossrate"
      }
    },
   "endpoint-cost-map": {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89"   : 0,
        "ipv4:198.51.100.34": 0,
        "ipv6:2000::1:2345:6789:abcd" : 0,
      }
    }
 }
```

3.5.4. Cost-Context Specification Considerations

"nominal": Typically packet loss rate does not have a nominal value, although some networks may specify zero losses.

"sla": It is RECOMMENDED that the "parameters" field of an "sla" packet loss rate provides a link ("link") to the SLA definition.

"import": There can be multiple sources to import packet loss rate. If the import is from [RFC8571] (by using unidirectional link loss), it is RECOMMENDED that "parameters" provides "protocol" as a field and "RFC8571" as the value; see Section 3.1.4 for discussions on summing up link metrics to obtain end-to-end metrics. If the import is from the IPPM framework, it is recommended that "parameters" provides "protocol" as a field and "ippm" as the value; see Section 4 of [I-D.ietf-ippm-initial-registry] for additional fields which can be specified for "ippm" in "parameters".

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" packet loss rate metric provides a link ("link") to a description of the "estimation" method.

4. Bandwidth Performance Metrics

This section introduces three bandwidth related metrics. Given a specified source to a specified destination, these metrics reflect the volume of traffic that the network can carry from the source to the destination.

4.1. Cost Metric: TCP Throughput (tput)

4.1.1. Base Identifier

The base identifier for this performance metric is "tput".

4.1.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of [RFC8259] Section 6. The number MUST be non-negative. The unit is bytes per second.

4.1.3. Intended Semantics and Use

Intended Semantics: To give the throughput of a TCP flow from the specified source to the specified destination. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 5: TCP throughput value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type": {"cost-mode" : "numerical",
                "cost-metric" : "tput"},
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
       "ipv4:192.0.2.89",
       "ipv4:198.51.100.34",
       "ipv6:2000::1:2345:6789:abcd"
    ]
  }
}

HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta": {
     "cost type": {
        "cost-mode": "numerical",
        "cost-metric":"tput"
    }
  }
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89"   : 256000,
      "ipv4:198.51.100.34": 128000,
      "ipv6:2000::1:2345:6789:abcd" : 428000,
  }
}
```

4.1.4. Cost-Context Specification Considerations

"nominal": Typically TCP throughput does not have a nominal value.

"sla": Typically TCP throughput does not have an SLA value.

"import": Typically there is not a routing protocol through which one can import TCP throughput. If the import is from the IPPM framework, it is recommended that "parameters" provides "protocol" as a field and "ippm" as the value; see Section 4 of [I-D.ietf-ippm-initial-registry] for additional fields which can be specified for "ippm" in "parameters".

"estimation": The exact estimation method is out of the scope of this document. See [Prophet] for a method to estimate TCP throughput. It is RECOMMENDED that the "parameters" field of an "estimation" TCP throughput metric provides a link ("link") to a description of the "estimation" method.

4.2. Cost Metric: Residue Bandwidth (bw-residue)

4.2.1. Base Identifier

The base identifier for this performance metric is "bw-residue".

4.2.2. Value Representation

The metric value type is a single 'JSONNumber' type value that is non-negative. The unit of measurement is bytes per second.

4.2.3. Intended Semantics and Use

Intended Semantics: To specify spatial and temporal residual bandwidth from the specified source and the specified destination. The value is calculated by subtracting tunnel reservations from Maximum Bandwidth (motivated from [RFC7810], Section 4.5). The spatial aggregation unit is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 7: bw-residue value on source-destination endpoint pairs

```
POST/ endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

  {
   "cost-type": { "cost-mode":   "numerical",
                  "cost-metric": "bw-residue"},
   "endpoints":  {
     "srcs": [ "ipv4 : 192.0.2.2" ],
     "dsts": [
       "ipv4:192.0.2.89",
       "ipv4:198.51.100.34",
       "ipv6:2000::1:2345:6789:abcd"
     ]
   }
  }

HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta": {
    "cost-type" {
      "cost-mode": "numerical",
      "cost-metric": "bw-residue"
    }
  },
  "endpoint-cost-map" {
    "ipv4:192.0.2.2" {
      "ipv4:192.0.2.89" :    0,
      "ipv4:198.51.100.34": 2000,
      "ipv6:2000::1:2345:6789:abcd":  5000,
    }
  }
}
```

4.2.4. Cost-Context Specification Considerations

"nominal": Typically residue bandwidth does not have a nominal value.

"sla": Typically residue bandwidth does not have an "sla" value.

"import": There can be multiple sources to import residue bandwidth. If the import is from [RFC8571] (by using unidirectional residue bandwidth), it is RECOMMENDED that "parameters" provides "protocol" as a field and "RFC8571" as the value. The server should be cognizant of issues when computing end-to-end summary statistics from link statistics. For example, the min of the end-to-end path residue bandwidth is the min of all links on the path.

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" residue bandwidth metric provides a link ("link") to a description of the "estimation" method.

4.3. Cost Metric: Maximum Reservable Bandwidth (bw-maxres)

4.3.1. Base Identifier

The base identifier for this performance metric is "bw-maxres".

4.3.2. Value Representation

The metric value type is a single 'JSONNumber' type value that is non-negative. The unit of measurement is bytes per second.

4.3.3. Intended Semantics and Use

Intended Semantics: To specify spatial and temporal maximum reservable bandwidth from the specified source to the specified destination. The value is corresponding to the maximum bandwidth that can be reserved (motivated from [RFC3630] Section 2.5.7). The spatial aggregation unit is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 6: bw-maxres value on source-destination endpoint pairs

```
POST/ endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

  {
    "cost-type" { "cost-mode":   "numerical",
                  "cost-metric": "bw-maxres"},
    "endpoints":  {
      "srcs": [ "ipv4 : 192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv6:2000::1:2345:6789:abcd"
      ]
    }
  }

HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta": {
    "cost-type": {
      "cost-mode":   "numerical",
      "cost-metric": "bw-maxres"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2" {
      "ipv4:192.0.2.89" :    0,
      "ipv4:198.51.100.34": 2000,
      "ipv6:2000::1:2345:6789:abcd":  5000,
    }
  }
}
```

4.3.4. Cost-Context Specification Considerations

"nominal": Typically maximum reservable bandwidth does not have a nominal value.

"sla": Typically maximum reservable bandwidth does not have an "sla" value.

"import": There can be multiple sources to import maximum reservable bandwidth. For example, Maximum reservable bandwidth is defined by IS-IS/OSPF TE, and measures the reservable bandwidth between two directly connected IS-IS neighbors or OSPF neighbors; see Section 3.5 of [RFC5305]. If the import is from [RFC8571] (by using unidirectional maximum reservable bandwidth), it is RECOMMENDED that "parameters" provides "protocol" as a field and "RFC8571" as the value.

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" maximum reservable bandwidth metric provides a link ("link") to a description of the "estimation" method.

5. Operational Considerations

The exact measurement infrastructure, measurement condition and computation algorithms can vary from different networks, and are outside the scope of this document. Both the ALTO server and the ALTO clients, however, need to be cognizant of the operational issues discussed below.

Also, the performance metrics specified in this document are similar, in that they may use similar data sources and have similar issues in their calculation. Hence, we specify common issues unless one metric has its unique challenges.

5.1. Source Considerations

The addition of the "cost-source" field is to solve a key issue: An ALTO server needs data sources to compute the cost metrics described in this document and an ALTO client needs to know the data sources to better interpret the values.

To avoid too fine-grained information, this document introduces "cost-source" to indicate only the high-level type of data sources: "estimation" or "sla", where "estimation" is a type of measurement data source and "sla" is a type that is more based on policy.

For estimation, for example, the ALTO server may use log servers or the OAM system as its data source as recommended by [RFC7971]. In particular, the cost metrics defined in this document can be computed using routing systems as the data sources.

5.2. Metric Timestamp Consideration

Despite the introduction of the additional cost-context information, there is not a built-in field to indicate the timestamps of the data used to compute a metric. To indicate this attribute, the ALTO server SHOULD return HTTP "Last-Modified", to indicate the freshness of the data used to compute the performance metrics.

If the ALTO client obtains updates through an incremental update mechanism (e.g., RFC editor: Fix the RFC number when available. [ALTO SSE]), the client SHOULD assume that the metric is computed using a snapshot at the time that is approximated by the receiving time.

5.3. Backward Compatibility Considerations

One potential issue introduced by the optional "cost-source" field is backward compatibility. Consider that an IRD which defines two cost-types with the same "cost-mode" and "cost-metric", but one with "cost-source" being "estimation" and the other being "sla". Then an ALTO client that is not aware of the extension will not be able to distinguish between these two types. A similar issue can arise even with a single cost-type which has "cost-source" being "sla", but the backward client will ignore this field and consider the metric estimation.

To address this issue, the only defined "routingcost" metric can be ONLY "estimation".

5.4. Computation Considerations

The metric values exposed by an ALTO server may result from additional processing on measurements from data sources to compute exposed metrics. This may involve data processing tasks such as aggregating the results across multiple systems, removing outliers, and creating additional statistics. There are two challenges on the computation of ALTO performance metrics.

5.4.1. Configuration Parameters Considerations

Performance metrics often depend on configuration parameters. For example, the value of packet loss rate depends on the measurement interval and varies over time. To handle this issue, an ALTO server may collect data on time periods covering the previous and current time or only collect data on present time. The ALTO server may further aggregate these data to provide an abstract and unified view that can be more useful to applications. To make the ALTO client better understand how to use these performance data, the ALTO server may provide the client with the validity period of the exposed metric values.

5.4.2. Availability Considerations

Applications value information relating to bandwidth availability whereas bandwidth related metrics can often be only measured at the link level. This document specifies a set of link-level bandwidth related values that may be exposed as such by an ALTO server. The server may also expose other metrics derived from their aggregation and having different levels of endpoint granularity, e.g., link endpoints or session endpoints. The metric specifications may also expose the utilized aggregation laws.

6. Security Considerations

The properties defined in this document present no security considerations beyond those in Section 15 of the base ALTO specification [RFC7285].

However concerns addressed in Sections "15.1 Authenticity and Integrity of ALTO Information", "15.2 Potential Undesirable Guidance from Authenticated ALTO Information" and "15.3 Confidentiality of ALTO Information" remain of utmost importance. Indeed, TE performance is a highly sensitive ISP information, therefore, sharing TE metric values in numerical mode requires full mutual confidence between the entities managing the ALTO Server and Client. Numerical TE performance information will most likely be distributed by ALTO Servers to Clients under strict and formal mutual trust agreements. On the other hand, ALTO Clients must be cognizant on the risks attached to such information that they would have acquired outside formal conditions of mutual trust.

7. IANA Considerations

IANA has created and now maintains the "ALTO Cost Metric Registry", listed in Section 14.2, Table 3 of [RFC7285]. This registry is located at <http://www.iana.org/assignments/alto-protocol/alto-protocol.xhtml#cost-metrics>. This document requests to add the following entries to "ALTO Cost Metric Registry".

Identifier	Intended Semantics
delay-ow	See Section 3.1
delay-rt	See Section 3.2
delay-variation	See Section 3.3
hopcount	See Section 3.4
lossrate	See Section 3.5
tput	See Section 4.1
bw-residue	See Section 4.2
bw-maxres	See Section 4.3

Since he This document requests the creation of the "ALTO Cost Source Registry" with the following currently defined values:

This document requests the creation of the "ALTO Cost Source Registry" with the following currently defined values:

Identifier	Intended Semantics
nominal	Values in nominal cases
sla	Values reflecting service level agreement
import	Values from a given protocol
estimation	Values by estimation

8. Acknowledgments

The authors of this document would also like to thank Brian Trammell, Haizhou Du, Kai Gao, Lili Liu, Geng Li, Danny Alex Lachos Perez for the reviews and comments.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <https://www.rfc-editor.org/info/rfc6390>.

[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>.

[RFC7971] Stiemerling, M., Kiesel, S., Scharf, M., Seidel, H., and S. Previdi, "Application-Layer Traffic Optimization (ALTO) Deployment Considerations", RFC 7971, DOI 10.17487/RFC7971, October 2016, <https://www.rfc-editor.org/info/rfc7971>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

9.2. Informative References

[Prometheus] Volz, J. and B. Rabenstein, "Prometheus: A Next-Generation Monitoring System", 2015.

[Prophet] Gao, K., Zhang, J., and YR. Yang, "Prophet: Fast, Accurate Throughput Prediction with Reactive Flows", ACM/IEEE Transactions on Networking July, 2020.

[RFC2679] Almes, G., Kalidindi, S., and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, DOI 10.17487/RFC2679, September 1999, <https://www.rfc-editor.org/info/rfc2679>.

[RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <https://www.rfc-editor.org/info/rfc2681>.

[RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <https://www.rfc-editor.org/info/rfc3393>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <https://www.rfc-editor.org/info/rfc5305>.

[RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <https://www.rfc-editor.org/info/rfc5357>.

[RFC6349] Constantine, B., Forget, G., Geib, R., and R. Schrage, "Framework for TCP Throughput Testing", RFC 6349, DOI 10.17487/RFC6349, August 2011, <https://www.rfc-editor.org/info/rfc6349>.

[RFC7680] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Loss Metric for IP Performance Metrics (IPPM)", STD 82, RFC 7680, DOI 10.17487/RFC7680, January 2016, <https://www.rfc-editor.org/info/rfc7680>.

[RFC7810] Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 7810, DOI 10.17487/RFC7810, May 2016, <https://www.rfc-editor.org/info/rfc7810>.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
CHINA

Email: bill.wu@huawei.com

Y. Richard Yang
Yale University
51 Prospect St
New Haven, CT 06520
USA

Email: yry@cs.yale.edu

Young Lee
Samsung
1700 Alma Drive, Suite 500
Plano, TX 75075
USA

Email: young.lee@gmail.com

Dhruv Dhody
Huawei
Leela Palace
Bangalore, Karnataka 560008
INDIA

Email: dhruv.ietf@gmail.com

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
Nozay 91460
FRANCE

Email: sabine.randriamasy@nokia-bell-labs.com

Luis Miguel Contreras Murillo
Telefonica
Madrid
SPAIN

Email: luismiguel.contrerasmurillo@telefonica.com

ALTO WG W. Roome
Internet-Draft S. Randriamasy
Intended status: Standards Track Nokia Bell Labs
Expires: January 14, 2021 Y. Yang
Yale University
J. Zhang
Tongji University
K. Gao
Sichuan University
July 13, 2020

Unified properties for the ALTO protocol
draft-ietf-alto-unified-props-new-12

Abstract

This document extends the Application-Layer Traffic Optimization (ALTO) Protocol [RFC7285] by generalizing the concept of "endpoint properties" to generic types of entities, and by presenting those properties as maps, similar to the network and cost maps in [RFC7285].

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

Copyright Notice

Table of Contents

1. Introduction

The ALTO protocol [RFC7285] introduces the concept of "properties" attached to "endpoint addresses", and defines the Endpoint Property Service (EPS) to allow ALTO clients to retrieve those properties. While useful, the EPS, as defined in [RFC7285], has at least three limitations.

First, the EPS allows properties to be associated with only endpoints which are identified by individual communication addresses like IPv4 and IPv6 addresses. It is reasonable to think that collections of endpoints, as defined by CIDRs [RFC4632] or PIDs, may also have properties. Furthermore, recent ALTO use cases show that properties of network flows [RFC7011] and routing elements [RFC7921] are also very useful. Since the EPS cannot be extended to those generic entities, new services, with new request and response messages, would have to be defined for them.

Second, the EPS only allows endpoints identified by global communication addresses. However, an endpoint address may be a local IP address or an anycast IP address which is also not globally unique. Additionally, a generic entity such as a PID may have an identifier that is not globally unique. For example, a PID identifier may be used in multiple network maps, where in each network map, this PID identifier points to a different set of addresses.

Third, the EPS is only defined as a POST-mode service. Clients must request the properties for an explicit set of endpoint addresses. By contrast, [RFC7285] defines a GET-mode cost map resource which returns all available costs, so a client can get a full set of costs once, and then process cost lookups without querying the ALTO server. [RFC7285] does not define a similar service for endpoint properties. At first, a map of endpoint properties might seem impractical, because it could require enumerating the property value for every possible endpoint. However, in practice, it is highly unlikely that properties will be defined for every endpoint address. It is much more likely that properties may be defined for only a subset of endpoint addresses, and the specification of properties uses an aggregation representation to allow enumeration. This is particularly true if blocks of endpoint addresses with a common prefix (e.g., a CIDR) have the same value for a property. Entities in other domains may very well allow aggregated representation and hence be enumerable as well.

To address the three limitations, this document specifies a protocol extension for defining and retrieving ALTO properties:

- o The first limitation is addressed by introducing a generic concept called ALTO Entity, which generalizes an endpoint and may represent a PID, a network element, a cell in a cellular network, an abstracted network element as defined in [REF path-vector], or other physical or logical objects used by ALTO. Each entity is included in a collection called an ALTO Entity Domain. Since each ALTO Entity Domain includes only one type of entities, each Entity Domain can be classified by the type of entities in it.

- o The second limitation is addressed by using resource-specific entity domains. A resource-specific entity domain contains entities that are defined and identified with respect to a given ALTO information resource, which provides scoping. For example, an entity domain containing PIDs is identified with respect to the network map in which these PIDs are defined. Likewise an entity domain containing local IP addresses may be defined with respect to a local network map.

- o The third limitation is addressed by defining two new types of ALTO information resources: Property Map, detailed in Section 7 and Filtered Property Map, detailed in Section 8. The former is a GET-mode resource that returns the property values for all entities in one or more entity domains, and is analogous to a network map or a cost map in [RFC7285]. The latter is a POST-mode resource that returns the values for a set of properties and entities requested by the client, and is analogous to a filtered network map or a filtered cost map.

The protocol extension defined in this document is extensible. New entity domain types can be defined without revising the specification defined in this document. Similarly, new cost metrics and new endpoint properties can be defined in other documents without revising the protocol specification defined in [RFC7285].

This document subsumes the Endpoint Property Service defined in [RFC7285], although that service may be retained for legacy clients (see Section 9).

This document assumes the reader is familiar with the base ALTO protocol defined in [RFC7285].

1.1. Terminology

TODO: TBC

- o Client: When starting with a capital "C", this term refers to an ALTO client.
- o Server: When starting with a capital "S", this term refers to an ALTO server.
- o TBC

2. Requirements Language

TODO: REAL RFC xrefs The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here. When the words appear in lower case, they are to be interpreted with their natural language meanings.

3. Basic Features of the Unified Property Extension

This section gives a high-level overview of the the basic features involved in ALTO Entity Property Maps. It assumes the reader is familiar with the ALTO protocol [RFC7285]. The purpose of this extension is to convey properties on objects that extend ALTO Endpoints and are called ALTO Entities, entities for short.

3.1. Entity

The concept of an ALTO Entity generalizes the concept of an ALTO Endpoint defined in Section 2.1 of [RFC7285]. An entity is an object that can be an endpoint that is defined by its network address, but can also be an object that has a defined mapping to a set of one or more network addresses or an object that is not even related to any network address. Thus, where as all endpoints are entities, not all entities are endpoints.

Examples of entities are:

- an ALTO endpoint, defined in [RFC7285], that represents an application or a host identified by a communication address (e.g., an IPv4 or IPv6 address) in a network,
- a PID, defined in [RFC7285], that has a provider defined human-readable identifier specified by an ALTO network map, which maps a PID to a set of ipv4 and ipv6 addresses,
- an autonomous system (AS), that has an AS number (ASN) as its identifier and maps to a set of ipv4 and ipv6 addresses,
- a country with a code as specified in [ISO3166-1], to which applications such as CDN providers associate properties and capabilities,
- a TCP/IP network flow, that is identified by a TCP/IP 5-Tuple specifying its source and destination addresses and port numbers and the utilized protocol,
- a routing element, that is specified in [RFC7921] and is associated with routing capabilities information,
- an abstract network element, that represents an abstraction of a network part such as a routable network node, one or more links, a network domain or their aggregation.

3.2. Entity Domain

An entity domain defines a set of entities of the same semantic type. An entity domain is characterized by its type and identified by its name.

In this document, an entity must be owned by exactly one entity domain name. An entity identifier must point to exactly one entity. If two entities in two different entity domains refer to the same physical or logical object, they are treated as different entities. For example, if an object has both an IPv4 and an IPv6 address, these two addresses will be treated as two entities, defined respectively in the "ipv4" and "ipv6" entity domains.

3.2.1. Entity Domain Type

The type of an entity domain type defines the semantics of a type of entity. Entity domain types can be defined in different documents. For example: the present document defines entity domain types "ipv4", "ipv6" and "pid" in sections Section 6.1 and Section 6.2. The entity domain type "ane", that defines Abstract Network Elements (ANEs), is introduced in [I-D.ietf-alto-path-vector]. The entity domain type that defines country codes is introduced in [draft-ietf-alto-cdni-request-routing-alto]. An entity domain type is expected to be registered at the IANA, as specified in section Section 12.2.2 and similarly to an ALTO address type.

3.2.2. Entity Domain Name

The name of an entity domain is defined in the scope of an ALTO server. An entity domain name can be identical to its relevant entity domain type in the following case: when the entities of an entity domain have an identifier that points to the same object throughout all the information resources of the Server that provide entity properties for this domain. For example, a domain of type "ipv4" that contains entities identified by a public IPv4 address can be named "ipv4" because its entities are uniquely identified by all the resources of the Server.

In some cases, a domain type and domain name must be different. Indeed, for some domain types, entities are defined relatively to a given information resource. As a consequence, entities in such domains may be defined in a resource handling this domain type but not in other resources handling this same domain type. Moreover, across different ALTO information resources handling a domain type, an entity identifier may point to different objects. This is the case for entities of domain type "pid". A PID is defined relatively to a network map. For example: an entity "mypid10" of domain type "pid" may be defined in a given network map resource and be undefined in other network maps, or may even map to a different set of endpoint addresses. In this case, naming an entity domain only by its type "pid" does not guarantee that its entities are owned by exactly one entity domain name. Section 4.2 and related of this document describe how a domain is uniquely identified by a name that associates the domain type and the related information resource.

3.3. Entity Property Type

An entity property defines a property of an entity. This is similar to the endpoint property defined in Section 7.1 of [RFC7285]. An entity property can convey either network-aware or network-agnostic information. Simularly to an entity domain, an entity property is characterized by its type and identified by its name. An entity property type is expected to be registered at the IANA, as specified in section Section 12.3.

For example:

- o an entity in the "ipv4" domain may have a property whose value is an Autonomous System (AS) number indicating the AS that owns this IPv4 address and another property named "countrycode" indicating a country code mapping to this address,

- o an entity identified by its country code in the "countrycode" domain, defined in [draft-ietf-alto-cdni-request-routing-alto] may have a property indicating what delivery protocol is used by a CDN,

- o an entity in the "netmap1.pid" domain may have a property that indicates the central geographical location of the endpoints it includes.

It should be noted that some identifiers may be used for both an entity domain type and a property type. For example:

- o the identifier "countrycode" may point to both the entity domain type "countrycode" and the property type "countrycode".

- o the identifier "pid" may point to both the entity domain type "pid" and the property type "pid".

Likewise, a same identifier may point to both a domain name and a property name.

3.4. New information resource and media type: ALTO Property Map

This document introduces a new ALTO information resource named Property Map. An ALTO property map provides a set of properties on one or more sets of entities. A property may apply to different entity domain types and names. For example, an ALTO property map may define the "ASN" property for both "ipv4" and "ipv6" entity domains.

The present extension also introduces a new media type.

This document uses the same definition of an information resource as Section 9.1 of [RFC7285]. ALTO uses media types to uniquely indicate the data format used to encode the content to be transmitted between an ALTO server and an ALTO client in the HTTP entity body. In the present case, an ALTO property map resource is defined by the media type "application/alto-propmap+json".

A Property Map can be queried as a GET-mode resource, thus conveying values of all properties on all entities indicated in its capabilities. It can also be queried as a POST-mode resource, thus conveying a selection of properties on a selection of entities.

4. Advanced Features of the Unified Property Extension

4.1. Entity Identifier and Entity Domain Name

In [RFC7285], an endpoint has an identifier which is explicitly associated with the "ipv4" or "ipv6" address domain. Examples are "ipv4:192.0.2.14" and "ipv6:2001:db8::12".

In this document, an entity must be owned by exactly one entity domain name and an entity identifier must point to exactly one entity. To ensure this, an entity identifier is explicitly attached to the name of its entity domain and an entity domain type characterizes the semantics and identifier format of its entities.

The encoding format of an entity identifier is further specified in Section 5.1.3 of this document.

For instance:

- o if an entity is an endpoint with example routable IPv4 address "192.0.2.14", its identifier is associated with domain name "ipv4" and is "ipv4:192.0.2.14",

- o if an entity is a PID named "mypid10" in network map resource "netmap2", its identifier is associated with domain name "netmap2.pid" and is "netmap2.pid:mypid10".

4.2. Resource-Specific Entity Domain Name

Some entities are defined and identified in a unique and global way. This is the case for instance for entities that are endpoints identified by a routable IPv4 or IPv6 address. The entity domain for such entities can be globally defined and named "ipv4" or "ipv6". Those entity domains are called resource-agnostic entity domains in this document, as they are not associated to any specific ALTO information resources.

Some other entities and entity types are only defined relatively to a given information resource. This is the case for entities of domain type "pid", that can only be understood with respect to the network map where they are defined. For example, a PID named "mypid10" may be defined to represent a set S1 of IP addresses in a network map resource named "netmap1". Another network map "netmap2" may use the same name "mypid10" and define it to represent another set S2 of IP addresses. The identifier "pid:mypid10" may thus point to different objects because the information on the originating information resource is lost.

To solve this ambiguity, the present extension introduces the concept of resources-specific entity domain. This concept applies to domain types where entities are defined relatively to a given information resource. It can also apply to entity domains that are defined locally, such as local networks of objects identified with a local IPv4 address.

In such cases, an entity domain type is explicitly associated with an identifier of the information resource where these entities are defined. Such an information resource is referred to as the "specific information resource". Using a resource-aware entity domain name, an ALTO property map can unambiguously identify distinct entity domains of the same type, on which entity properties may be queried. Example resource-specific entity domain names may look like: "netmap1.pid" or "netmap2.pid". Thus, a name association such as "netmap1.pid:mypid10" and "netmap2.pid:mypid10" allows to distinguish the two abovementioned PIDs that are both named "mypid10" but in two different resources, "netmap1" and "netmap2".

An information resource is defined in the scope of an ALTO Server and so is an entity domain name. The format of a resource-specific entity domain name is further specified in Section 5.1.2.

4.3. Resource-Specific Entity Property Value

Like entity domains, some types of properties are defined relatively to an information resource. That is, an entity may have a property of a given type, whose values are associated to different information resources.

For example, suppose entity "192.0.2.34" defined in the "ipv4" domain has a property of type "pid", whose value is the PID to which address "192.0.2.34" is attached in a network map. The mapping of network addresses to PIDs is specific to a network map and probably different from one network map resource to another one. So that if a property "pid" is defined for entity "192.0.2.34" in two different network maps "netmap1" and "netmap2", the value for this property will likely be different value in "netmap1" and "netmap2".

To support information resource dependent property values, this document uses the same approach as in Section 10.8.1 of [RFC7285] entitled "Resource-Specific Endpoint Properties". When a property value depends on a given information resource, the name of this property must be explicitly associated with the information resource that defines it.

For example, the property "pid" queried on entity "ipv4:192.0.2.34" and defined in both "netmap1" and "netmap2", can be named "netmap1.pid" and "netmap2.pid". This allows a Client to get a property of the same type but defined in different information resources with a single query. Specifications on the property name format are provided in Section 5.2.

4.4. Entity Hierarchy and Property Inheritance

For some domain types, entities can be grouped in a set and be defined by the identifier of this set. This is the case for domain types "ipv4" and "ipv6", where individual Internet addresses can be grouped in blocks. When a same property value applies to a whole set, a Server can define a property for the identifier of this set instead of enumerating all the entities and their properties. This allows substantial reduction of transmission payload both for the Server and the Client. For example, all the entities included in the set defined by the address block "ipv6:2001:db8::1/64" share the same properties and values defined for this block.

Additionally, entity sets sometimes are related by inclusion, hierarchy or other relations. This allows defining inheritance rules for entity properties that propagate properties among related entity sets. The Server and the Client can use these inheritance rules for further payload savings. Entity hierarchy and property inheritance rules are specified in the documents that define the applicable domain types. The present document defines these rules for the "ipv4" and "ipv6" domain types.

This document introduces, for applicable domain types, "Entity Property Inheritance rules", with the following concepts: Entity Hierarchy, Property Inheritance and Property Value Unicity. A detailed specification of entity hierarchy and property inheritance rules is provided in Section 5.1.4.

4.4.1. Entity Hierarchy

An entity domain may allow using a single identifier to identify a set of individual entities. For example, a CIDR block can be used to identify a set of IPv4 or IPv6 entities. A CIDR block is called a hierarchical entity identifier, as it can reflect inclusion relations among entity sets. For example, the CIDR "ipv4:192.0.1.0/24" includes all the individual ipv4 entities identified by the CIDR "ipv4:192.0.1.0/26".

4.4.2. Property Inheritance

A property may be defined for a hierarchical entity identifier while it may be undefined for individual entities covered by this identifier. In this case these individual entities inherit the property value defined for the identifier that covers them. For example, suppose a property map defines the ASN property only for the hierarchical entity identifier "ipv4:192.0.1.0/24" but not for individual entities in this block. Suppose also that inheritance rules are specified for CIDR blocks in the "ipv4" domain type. When receiving this property map, a Client can infer that entity "ipv4:192.0.1.1" inherits the "ASN" property value of block "ipv4:192.0.1.0/24" because the address "ipv4:192.0.1.1" is included by the CIDR block "ipv4:192.0.1.0/24".

Property value inheritance rules also apply among entity sets. A property map may define values for an entity set belonging to a hierarchy but not for "sub" sets that are covered by this set identifier. In this case, inheritance rules must specify how entities in "sub" sets inherit property values from their "super" set. For instance, if the "ASN" property is defined only for the entity set identified by block "ipv4:192.0.1.0/24", the entity set identified by "ipv4:192.0.1.0/30" and thus included in the former set, may inherit the "ASN" property values from set "ipv4:192.0.1.0/24".

4.4.3. Property Value Unicity

The inheritance rules must ensure that an entity belonging to a hierarchical set of entities inherits no more than one property value. Indeed, a property map may define a property on a hierarchy of entity sets that inherit property values from one or more including sets in the upper levels. On the other hand, a property value, defined at a lower level of the hierarchy may be different from the value defined at an upper level. In such a case, a set in the lower level of the hierarchy may potentially end up with different values. This may be the case for address blocs with increasing prefix length, on which a property value gets increasingly accurate and thus may differ. For example, a fictitious property such as "geo-location" or "average transfer volume" may be defined at a progressively finer grain for entity sets defined with progressively longer CIDR prefixes. It seems more interesting to have property values of progressively higher accuracy. A unicity rule, applied to the entity domain type must specify an arbitration rule among the different property values for an entity.

4.5. Supported Properties on Entity Domains in Property Map Capabilities

A property type is not necessarily applicable to any domain type, or an ALTO Server may just not provide a property on all applicable domains. For instance, a property type reflecting link bandwidth is likely not defined on entities of a domain of type "country-code". Therefore an ALTO server providing Property Maps specifies the properties that can be queried on the different entity domains it supports.

This document explains how the IRD capabilities of a Property Map resource unambiguously expose what properties a Client can query on a given entity domain.

- o a field named "mappings" lists the names of the entity domains supported by the Property Map,

- o for each listed entity domain, a list of the names of the applicable properties is provided.

An example is provided in Section 10.4. The "mappings" field associates entity domains and properties that can be resource-agnostic or resource-specific. This allows a Client to formulate compact and unambiguous entity property queries, possibly relating to one or more information resources. In particular:

- o it avoids a Client to query a property on entity domains on which it is not defined,

- o it allows a Client to query, for an entity E, values for a property P that are defined in different information resources,

- o it allows a Client to query a property P on entities that are defined in different information resources.

Further specifications are provided in Section 7.4.

4.6. Defining Information Resource

A Client willing to query properties on entities belonging to a domain needs to know how to retrieve these entities. To this end, he Client can look up the "mappings" field exposed in IRD capabilities of a property map, see Section 4.5. This field, in its keys, exposes all the entity domains supported by the property map. The syntax of the entity domain identifier specified in Section 5.1.2 allows the client to infer whether the entity domain is resource-specific or not. The Client can extract, if applicable, the identifier of the specific resource, query the resource and retrieve the entities. For example:

- o an entity domain named "netmap1.ipv4" includes the IPv4 addresses that appear in the "ipv4" field of the endpoint address group of each PID in the network map "netmap1", and that cannot be recognized outside "netmap1", for instance because these are local non routable addresses,

- o an entity domain named "netmap1.pid" includes the PIDs listed in network map "netmap1".

- o an entity domain named "ipv4" is resource-agnostic and covers all the routable IPv4 addresses.

Besides, it is also necessary to inform a Client about which associations of specific resources and entity domain types are allowed, because it is not possible to prevent a Server from exposing inappropriate associations. An informed Client will just ignore inappropriate associations exposed by a Server and avoid error-prone transactions with the Server.

For example, the association "costmap3.pid" is not allowed for the following reason: although a cost map exposes PID identifiers, it does not define them, that is, the set of addresses included in this PID. Neither does a cost map list all the PIDs on which properties can be queried, because a cost map only exposes PID pairs on which a queried cost type is defined. The resource "costmap3" therefore does not enable a Client to extract information on the existing PID entities or on the addresses they contain.

Instead, the cost map uses a network map, that lists all the PIDs used in a cost map, together with the addresses contained by the PIDs. This network map is qualified in this document as the Defining Information Resource for the entity domain "pid" and this concept is explained in Section 4.6.1.

4.6.1. Defining Information Resource and Media Type

For the reasons explained in the previous section, this document indroduces the concept of defining information resource and media type.

A defining information resource for an entity domain D is the information resource where entities of D are defined. That is, all the information on the entities of D can be retrieved in this resource. This concept applies to resource specific domains. This is useful for entity domain types that are by essence domain-specific, such as "pid" and "ane" domain types. It is also useful for resource-specific entity domains constructed from resource-agnostic domain types, such as for example, network map specific domains of local IPv4 addresses.

The defining information resource of an entity domain D has the following specificities:

- it has an entry in the IRD,

- it defines the entities of D,

- it does not use another information resource that defines these entities,

- it defines and exposes entity identifiers that are all persistent.

- its media type is unique and equal to the one that is specified for the defining information resource of an entity domain type.

A fundamental attribute of a defining information resource is its media type. There is a unique association of an entity domain type with the media type of its defining information resources. If an entity domain type allows defining information resources, their media type is specified in the document that defines this entity domain type and in the document that requests the registration of this domain type at the IANA.

When the Client wants to use a resource-specific entity domain, it needs to be cognizant of the media-type of its defining information resource. If the Server exposes resources a resource specific entity domain with a non compliant media type for the domain type, the Client can avoid transaction errors by ignoring them.

4.6.2. Examples of specific resources media-types

Here are some examples of specific information resources types associated to entity domain types and their media type.

- For entity domain type "pid": the media type of the specific resource is "application/alto-networkmap+json", because PIDs are defined in network map resources.

- For entity domain types "ipv4" and "ipv6": the media type of the specific resource is "application/alto-networkmap+json", because IPv4 and IPv6 addresses covered by the Server are defined in network map resources.

- o For entity domain type "ane", defined in [I-D.ietf-alto-path-vector]: a specific property map resource can be associated to ANEs that have a persistent identifier and have an entry in the IRD. ANEs that have a persistent identifier are defined in a property map that is indicated in the IRD, therefore the media type associated with entity domain type "ane" is "application/alto-propmap+json".

4.7. Defining Information Resource for Resource-Specific Property Values

As explained in Section 4.3, a property type may take values that are resource specific. This is the case for example for property type "pid", whose values are by essence defined relatively to a specific network map. The PID value for an IPv4 address differ in different network maps or not be defined for some of them. Property values may be specific to different types of information resources. For example: the value for property "pid" is specific to a network map. The value for property type "cdnifci-capab" is specific to the information resource "cdnifci-map", defined in [draft-ietf-alto-cdni-request-routing-alto], while network maps do not define property "fci-capability" for ipv4 addresses and a cdnifci-map does not define "pid" values for IPv4 addresses.

Thus, similarly to resource specific entity domains, the Client needs to be aware of aware of appropriate associations of information resource and property types.

4.7.1. Examples of defining resources media-types for properties

Here are some examples of specific information resources types associated to entity property types and their media type.

- o For property type "pid": the media type of the specific resource is "application/alto-networkmap+json", because PIDs are defined in network map resources.

- o For property type "cdni-fci-capability": the media type of the specific resource is "application/alto-cdnifci+json"

5. Protocol Specification: Basic Data Type

5.1. Entity Domain

5.1.1. Entity Domain Type

An entity domain has a type, which is uniquely identified by a string that MUST be no more than 64 characters, and MUST NOT contain characters other than US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A, and U+0061-U+007A), hyphen ("-", U+002D), and low line ("_", U+005F). The ?.? separator MUST NOT be used unless specifically indicated in a further extension document.

For example, the strings "ipv4", "ipv6", and "pid" are valid entity domain types. "ipv4.anycast" and "pid.local" are invalid.

The type EntityDomainType is used in this document to denote a JSON string meeting the preceding requirement.

An entity domain type defines the semantics of a type of entity, indenpendently of any specifying resource. Each entity domain type MUST be registered with the IANA. The format of the entity identifiers (see Section 5.1.3) in that type of entity domains, as well as any hierarchical or inheritance rules (see Section 5.1.4) for those entities, MUST be specified at the same time.

5.1.2. Entity Domain Name

This document distinguishes three categories of entity domains: resource-specific entity domains, resource-agnostic entity domains and self-defined entity domains. Their entity domain names are constructed as specified in the following sub-sections.

Each entity domain is identified by a unique entity domain name which is a string of the following format:

```
EntityDomainName ::= [ [ ResourceID ] '.' ] EntityDomainType
```

Where the presence and construction of component:

```
"[ [ ResourceID ] '.' ]"
```

depends on the category of entity domain.

Note that the '.' separator is not allowed in EntityDomainType and hence there is no ambiguity on whether an entity domain name refers to a resource-agnostic entity domain or a resource-specific entity domain.

Note also that the resource ID format definition in Section 10.1 of [RFC7285] specifies that: "the '.' separator is reserved for future use and MUST NOT be used unless specifically indicated in this document, or an extension document". The present extension keeps the format specification of [RFC7285], hence the '.' separator MUST NOT be used in an information resources ID.

5.1.2.1. Resource-specific Entity Domain

A resource-specific entity domain is identified by an entity domain name constructed as follows. It MUST start with a resource ID using the ResourceID type defined in Section 10.2 of [RFC7285], followed by the '.' separator (U+002E), followed by an string of the type EntityDomainType specified in Section 5.1.1.

For example, if an ALTO server provides two network maps "netmap-1" and "netmap-2", these network maps can define two resource-specific domains of type "pid", respectively identified by "netmap-1.pid" and "netmap-2.pid".

5.1.2.2. Resource-agnostic Entity Domain

A resource-agnostic entity domain contains entities that are identified independently of any information resource. Hence, the identifier of a resource-agnostic entity domain is simply the identifier of its entity domain type. For example, "ipv4" and "ipv6" identify the two resource-agnostic Internet address entity domains defined in Section 6.1.

5.1.2.3. Self-defined Entity Domain

A property map can define properties on entities that are neither resource-specific nor resource-agnostic but are instead defined within the property map itself. This may be the case when an ALTO Server provides information on a set of entities that is specific to this property map would not be relevant for another one and that does not depend on a specific resource.

For example: a specialised property map may define a domain of type "ane", defined in [I-D.ietf-alto-path-vector], that contains a set of ANEs with a persistent identifier that are relevant only for this property map.

In this case, the entity domain is qualified as "self-defined". The identifier of a self-defined entity domain can be of the format:

```
        EntityDomainName ::= .EntityDomainType
```

where '.' indicates that the entity domain only exists within the property map resource using it.

A self-defined entity domain can be viewed as a particular case of resource-specific entity domain, where the specific resource is the current resource that uses this entity domain. In that case, for the sake of simplification, the component "ResourceID" SHOULD be omitted in its entity domain name.

5.1.3. Entity Identifier

Entities in an entity domain are identified by entity identifiers (EntityID) of the following format:

```
EntityID ::= EntityDomainName ':' DomainTypeSpecificEntityID
```

Examples from the Internet address entity domains include individual IP addresses such as "net1.ipv4:192.0.2.14" and "net1.ipv6:2001:db8::12", as well as address blocks such as "net1.ipv4:192.0.2.0/26" and "net1.ipv6:2001:db8::1/48".

The format of the second part of an entity identifier depends on the entity domain type, and MUST be specified when defining a new entity domain type and registering it with the IANA. Identifiers MAY be hierarchical, and properties MAY be inherited based on that hierarchy. The rules defining any hierarchy or inheritance MUST be defined when the entity domain type is registered.

The type EntityID is used in this document to denote a JSON string representing an entity identifier in this format.

Note that two entity identifiers with different valid textual representations may refer to the same entity, for a given entity domain. For example, the strings "net1.ipv6:2001:db8::1" and "net1.ipv6:2001:db8:0:0:0:0:0:1" refer to the same entity in the "ipv6" entity domain.

5.1.4. Hierarchy and Inheritance

To simplify the representation, some types of entity domains allow the ALTO Client and Server to use a hierarchical entity identifier format to represent a block of individual entities. For instance, in an IPv4 domain "net1.ipv4", a CIDR "net1.ipv4:192.0.2.0/26" covers 64 individual IPv4 entities. In this case, the corresponding property inheritance rule MUST be defined for the entity domain type. The hierarchy and inheritance rule MUST have no ambiguity.

5.2. Entity Property

Each entity property has a type to indicate the encoding and the semantics of the value of this entity property, and has a name to identify it.

5.2.1. Entity Property Type

The type EntityPropertyType is used in this document to indicate a string denoting an entity property type. The string MUST be no more than 32 characters, and it MUST NOT contain characters other than US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A, and U+0061-U+007A), the hyphen ("-", U+002D), the colon (":", U+003A), or the low line ('_', U+005F). Note that the ?.? separator is not allowed because it is reserved to separate an entity property type and an information resource identifier when an entity property is resource-specific.

Each entity property type MUST be registered with the IANA. The intended semantics of the entity property type MUST be specified at the same time.

Identifiers prefixed with "priv:" are reserved for Private Use [RFC5226] without a need to register with IANA. All other identifiers for entity property types appearing in an HTTP request or response with an "application/alto-*" media type MUST be registered in the "ALTO Entity Property Type Registry", defined in Section 12.3. For an entity property identifier with the "priv:" prefix, an additional string (e.g., company identifier or random string) MUST follow (i.e., "priv:" only is not a valid endpoint property identifier) to reduce potential collisions.

To distinguish with the endpoint property type, the entity property type has the following features.

- o Some entity property types may be applicable to entities in only particular types of entity domains, not all. For example, the "pid" property is not applicable to entities in a "pid" typed entity domain, but is applicable to entities in the "ipv4" or "ipv6" domains.

- o The intended semantics of the value of an entity property may also depend on the entity domain type of this entity. For example, suppose that the "geo-location" property is defined as the coordinates of a point, encoded as (say) "latitude longitude [altitude]." When applied to an entity that represents a specific host computer, identified by an address in the "ipv4" or "ipv6" entity domain, the property defines the host's location. However, when applied to an entity in a "pid" domain, the property would indicate the location of the center of all hosts in this "pid" entity.

5.2.2. Entity Property Name

Each entity property is identified by an entity property name, which is a string of the following format:

```
EntityPropertyName ::= [ ResourceID ] '.' EntityPropertyType
```

Similar to the endpoint property type defined in Section 10.8 of [RFC7285], each entity property may be defined by either the property map itself (self-defined) or some other specific information resource (resource-specific).

The entity property name of a resource-specific entity property starts with a string of the type ResourceID defined in [RFC7285], followed by the "." separator (U+002E) and a EntityDomainType typed string. For example, the "pid" properties of an "ipv4" entity defined by two different maps "net-map-1" and "net-map-2" are identified by "net-map-1.pid" and "net-map-2.pid" respectively.

When the associated information resource of the entity property is the current information resource itself, the ResourceID in the property name SHOULD be ignored. For example, the ".asn" property of an "ipv4" entity indicates the AS number of the AS which this IPv4 address is owned by.

5.2.3. Format for Entity Property Value

[RFC7285] in Section 11.4.1.6, specifies that an implementation of the Endpoint Property Service specified in [RFC7285] SHOULD assume that the property value is a JSONString and fail to parse if it is not. The present document first, extends the property service to Entities. Second it extends the format of a property value by allowing it to be a JSONValue instead of just a JSONString.

6. Entity Domain Types Defined in this Document

This document requires the definition of each entity domain type MUST include (1) the entity domain type name and (2) domain-specific entity identifiers, and MAY include (3) hierarchy and inheritance semantics optionally. This document defines three initial entity domain types as follows.

6.1. Internet Address Domain Types

The document defines two entity domain types (IPv4 and IPv6) for Internet addresses. Both types are resource-agnostic entity domain types and hence define corresponding resource-agnostic entity domains as well. Since the two domains use the same hierarchy and inheritance semantics, we define the semantics together, instead of repeating for each.

6.1.1. IPv4 Domain

6.1.1.1. Entity Domain Type

ipv4

6.1.1.2. Domain-Specific Entity Identifiers

Individual addresses are strings as specified by the IPv4Addresses rule of Section 3.2.2 of [RFC3986]; Hierarchical addresses are prefix-match strings as specified in Section 3.1 of [RFC4632]. To define properties, an individual Internet address and the corresponding full-length prefix are considered aliases for the same entity. Thus "ipv4:192.0.2.0" and "ipv4:192.0.2.0/32" are equivalent.

6.1.2. IPv6 Domain

6.1.2.1. Entity Domain Type

ipv6

6.1.2.2. Domain-Specific Entity Identifiers

Individual addresses are strings as specified by Section 4 of [RFC5952]; Hierarchical addresses are prefix-match strings as specified in Section 7 of [RFC5952]. To define properties, an individual Internet address and the corresponding 128-bit prefix are considered aliases for the same entity. That is, "ipv6:2001:db8::1" and "ipv6:2001:db8::1/128" are equivalent, and have the same set of properties.

6.1.3. Hierarchy and Inheritance of Internet Address Domains

Both Internet address domains allow property values to be inherited. Specifically, if a property P is not defined for a specific Internet address I, but P is defined for a a hierarchical Internet address C which prefix-matches I, then the address I inherits the value of P defined for the hierarchical address C. If more than one such hierarchical addresses define a value for P, I inherits the value of P in the hierarchical address with the longest prefix. Note that this longest prefix rule ensures no multiple inheritances, and hence no ambiguity.

Hierarchical addresses can also inherit properties: if a property P is not defined for the hierarchical address C, but is defined for another hierarchical address C’ which covers all IP addresses in C, and C’ has a shorter prefix length than C, then C MAY inherits the property from C’. If there are multiple such hierarchical addresses like C’, C MUST inherit from the hierarchical address having the longest prefix length.

As an example, suppose that a server defines a property P for the following entities:

```
ipv4:192.0.2.0/26: P=v1
ipv4:192.0.2.0/28: P=v2
ipv4:192.0.2.0/30: P=v3
ipv4:192.0.2.0:    P=v4
```

Figure 1: Defined Property Values.

Then the following entities have the indicated values:

```
ipv4:192.0.2.0:    P=v4
ipv4:192.0.2.1:    P=v3
ipv4:192.0.2.16:   P=v1
ipv4:192.0.2.32:   P=v1
ipv4:192.0.2.64:   (not defined)
ipv4:192.0.2.0/32: P=v4
ipv4:192.0.2.0/31: P=v3
ipv4:192.0.2.0/29: P=v2
ipv4:192.0.2.0/27: P=v1
ipv4:192.0.2.0/25: (not defined)
```

Figure 2: Inherited Property Values.

An ALTO server MAY explicitly indicate a property as not having a value for a particular entity. That is, a server MAY say that property P of entity X is "defined to have no value", instead of "undefined". To indicate "no value", a server MAY perform different behaviours:

- o If that entity would inherit a value for that property, then the ALTO server MUST return a "null" value for that property. In this case, the ALTO client MUST recognize a "null" value as "no value" and "do not apply the inheritance rules for this property."

- If the entity would not inherit a value, then the ALTO server MAY return "null" or just omit the property. In this case, the ALTO client cannot infer the value for this property of this entity from the Inheritance rules. So the client MUST interpret that this property has no value.

If the ALTO server does not define any properties for an entity, then the server MAY omit that entity from the response.

6.1.4. Defining Information Resource Media Type for domain types IPv4 and IPv6

Entity domain types "ipv4" and "ipv6" both allow to define resource specific entity domains. When resource specific domains are defined with entities of domain type "ipv4" or "ipv6", the defining information resource for an entity domain of type "ipv4" or "ipv6" MUST be a Network Map. The media type of a defining information resource is therefore:

application/alto-networkmap+json

6.2. PID Domain

The PID domain associates property values with the PIDs in a network map. Accordingly, this entity domain always depends on a network map.

6.2.1. Entity Domain Type

pid

6.2.2. Domain-Specific Entity Identifiers

The entity identifiers are the PID names of the associated network map.

6.2.3. Hierarchy and Inheritance

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

6.2.4. Defining Information Resource Media Type for Domain Type PID

The entity domain type "pid" allows to define resource specific entity domains. When resource specific domains are defined with entities of domain type "pid", the defining information resource for entity domain type "pid" MUST be a Network Map. The media type of a defining information resource is therefore:

application/alto-networkmap+json

6.2.5. Relationship To Internet Addresses Domains

The PID domain and the Internet address domains are completely independent; the properties associated with a PID have no relation to the properties associated with the prefixes or endpoint addresses in that PID. An ALTO server MAY choose to assign some or all properties of a PID to the prefixes in that PID.

For example, suppose "PID1" consists of the prefix "ipv4:192.0.2.0/24", and has the property "P" with value "v1". The Internet address entities "ipv4:192.0.2.0" and "ipv4:192.0.2.0/24" in the IPv4 domain MAY have a value for the property "P", and if they do, it is not necessarily "v1".

6.3. Internet Address Properties vs. PID Properties

Because the Internet address and PID domains are completely separate, the question may arise as to which entity domain is the best for a property. In general, the Internet address domains are RECOMMENDED for properties that are closely related to the Internet address, or are associated with, and inherited through, hierarchical addresses.

The PID domain is RECOMMENDED for properties that arise from the definition of the PID, rather than from the Internet address prefixes in that PID.

For example, because Internet addresses are allocated to service providers by blocks of prefixes, an "ISP" property would be best associated with the Internet address domain. On the other hand, a property that explains why a PID was formed, or how it relates to a provider's network, would best be associated with the PID domain.

7. Property Map

A property map returns the properties defined for all entities in one or more domains, e.g., the "location" property of entities in "pid" domain, and the "ASN" property of entities in "ipv4" and "ipv6" domains.

Section 10.5 gives an example of a property map request and its response.

7.1. Media Type

The media type of a property map is "application/alto-propmap+json".

7.2. HTTP Method

The property map is requested using the HTTP GET method.

7.3. Accept Input Parameters

None.

7.4. Capabilities

The capabilities are defined by an object of type PropertyMapCapabilities:

```
object {
  EntityPropertyMapping mappings;
} PropertyMapCapabilities;

object-map {
  EntityDomainName -> EntityPropertyName<1..*>;
} EntityPropertyMapping
```

with fields:

mappings: A JSON object whose keys are names of entity domains and values are the supported entity properties of the corresponding entity domains.

7.5. Uses

The "uses" field of a property map resource in an IRD entry specifies dependent resources of this property map. It is an array of the resource ID(s) of the resource(s).

7.6. Response

If the entity domains in this property map depend on other resources, the "dependent-vtags" field in the "meta" field of the response MUST be an array that includes the version tags of those resources, and the order MUST be consistent with the "uses" field of this property map resource. The data component of a property map response is named "property-map", which is a JSON object of type PropertyMapData, where:

```
object {
  PropertyMapData property-map;
} InfoResourceProperties : ResponseEntityBase;

object-map {
  EntityID -> EntityProps;
} PropertyMapData;

object {
  EntityPropertyName -> JSONValue;
} EntityProps;
```

The ResponseEntityBase type is defined in Section 8.4 of [RFC7285].

Specifically, a PropertyMapData object has one member for each entity in the property map. The entity's properties are encoded in the corresponding EntityProps object. EntityProps encodes one name/value pair for each property, where the property names are encoded as strings of type PropertyName. A protocol implementation SHOULD assume that the property value is either a JSONString or a JSON "null" value, and fail to parse if it is not, unless the implementation is using an extension to this document that indicates when and how property values of other data types are signaled.

For each entity in the property map:

o If the entity is in a resource-specific entity domain, the ALTO server SHOULD only return self-defined properties and resource-specific properties which depend on the same resource as the entity does. The ALTO client SHOULD ignore the resource-specific property in this entity if their mapping is not registered in the ALTO Resource Entity Property Transfer Registry of the type of the corresponding resource.

o If the entity is in a shared entity domain, the ALTO server SHOULD return self-defined properties and all resource-specific properties defined for all resource-specific entities which have the same domain-specific entity identifier as this entity does.

For efficiency, the ALTO server SHOULD omit property values that are inherited rather than explicitly defined; if a client needs inherited values, the client SHOULD use the entity domain's inheritance rules to deduce those values.

8. Filtered Property Map

A filtered property map returns the values of a set of properties for a set of entities selected by the client.

Section 10.6, Section 10.7, Section 10.8 and Section 10.9 give examples of filtered property map requests and responses.

8.1. Media Type

The media type of a property map resource is "application/alto-propmap+json".

8.2. HTTP Method

The filtered property map is requested using the HTTP POST method.

8.3. Accept Input Parameters

The input parameters for a filtered property map request 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-propmapparams+json", which is a JSON object of type ReqFilteredPropertyMap:

```
object {
  EntityID             entities<1..*>;
  EntityPropertyName   properties<1..*>;
} ReqFilteredPropertyMap;
```

with fields:

entities: List of entity identifiers for which the specified properties are to be returned. The ALTO server MUST interpret entries appearing multiple times as if they appeared only once. The domain of each entity MUST be included in the list of entity domains in this resource's "capabilities" field (see Section 8.4).

properties: List of properties to be returned for each entity. Each specified property MUST be included in the list of properties in this resource's "capabilities" field (see Section 8.4). The ALTO server MUST interpret entries appearing multiple times as if they appeared only once.

Note that the "entities" and "properties" fields MUST have at least one entry each.

8.4. Capabilities

The capabilities are defined by an object of type PropertyMapCapabilities, as defined in Section 7.4.

8.5. Uses

Same to the "uses" field of the Property Map resource (see Section 7.5).

8.6. Response

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

Specifically, a filtered property map request can be invalid as follows:

- An entity identifier in "entities" in the request is invalid if:

  * The domain of this entity is not defined in the "entity-domains" capability of this resource in the IRD;

  * The entity identifier is an invalid identifier in the entity domain.

  A valid entity identifier is never an error, even if this filtered property map resource does not define any properties for it.

  If an entity identifier in "entities" in the 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 entity identifier.

- A property name in "properties" in the request is invalid if this property name is not defined in the "properties" capability of this resource in the IRD.

  It is not an error that a filtered property map resource does not define a requested property's value for a particular entity. In this case, the ALTO server MUST omit that property from the response for that endpoint.

  If a property name in "properties" in the request is invalid, the ALTO server MUST return an "E_INVALID_FIELD_VALUE" error defined in Section 8.5.2 of [RFC7285]. The "value" field of the error message SHOULD indicate the property name.

The response to a valid request is the same as for the Property Map (see Section 7.6), except that:

- If the requested entities include entities in the shared entity domain, the "dependent-vtags" field in its "meta" field MUST include version tags of all dependent resources appearing in the "uses" field.

- If the requested entities only include entities in resource-specific entity domains, the "dependent-vtags" field in its "meta" field MUST include version tags of resources which requested resource-specific entity domains and requested resource-specific properties are dependent on.

- The response only includes the entities and properties requested by the client. If an entity in the request is identified by a hierarchical identifier (e.g., a "ipv4" or "ipv6" prefix), the response MUST cover properties for all identifiers in this hierarchical identifier.

It is important that the filtered property map response MUST include all inherited property values for the requested entities and all the entities which are able to inherit property values from them. To achieve this goal, the ALTO server MAY follow three rules:

- If a property for a requested entity is inherited from another entity not included in the request, the response SHOULD include this property for the requested entity. For example, A full property map may skip a property P for an entity A (e.g., ipv4:192.0.2.0/31) if P can be derived using inheritance from another entity B (e.g., ipv4:192.0.2.0/30). A filtered property map request may include only A but not B. In such a case, the property P SHOULD be included in the response for A.

- If there are entities covered by a requested entity but having different values for the requested properties, the response SHOULD include all those entities and the different property values for them. For example, considering a request for property P of entity A (e.g., ipv4:192.0.2.0/31), if P has value v1 for A1=ipv4:192.0.2.0/32 and v2 for A2=ipv4:192.0.2.1/32, then, the response SHOULD include A1 and A2.

- If an entity in the response is already covered by some other entities in the same response, it SHOULD be removed from the response for compactness. For example, in the previous example, the entity A=ipv4:192.0.2.0/31 SHOULD be removed because A1 and A2 cover all the addresses in A.

An ALTO client should be aware that the entities in the response MAY be different from the entities in its request.

8.7. Entity property type defined in this document

This document defines the entity property type "pid"

The intended semantics are the same as in [RFC7285]

The defining information resource for property type MUST be a network map.

The media type of a defining information resource is therefore:

application/alto-networkmap+json

This document requests a IANA registration for this property

9. Impact on Legacy ALTO Servers and ALTO Clients

9.1. Impact on Endpoint Property Service

Since the property map and the filtered property map defined in this document provide the functionality of the Endpoint Property Service (EPS) defined in Section 11.4 of [RFC7285], it is RECOMMENDED that the EPS be deprecated in favor of Property Map and Filtered Property Map. However, ALTO servers MAY provide an EPS for the benefit of legacy clients.

9.2. Impact on Resource-Specific Properties

Section 10.8 of [RFC7285] defines two categories of endpoint properties: "resource-specific" and "global". Resource-specific property names are prefixed with the ID of the resource they depend upon, while global property names have no such prefix. The property map and the filtered property map defined in this document defines the similar categories for entity properties. The difference is that there is no "global" entity properties but the "self-defined" entity properties as the special case of the "resource-specific" entity properties instead.

9.3. Impact on Other Properties

In general, there should be little or no impact on other previously defined properties. The only consideration is that properties can now be defined on hierarchical entity identifiers, rather than just individual entity identifiers, which might change the semantics of a property.

10. Examples

10.1. Network Map

The examples in this section use a very simple default network map:

```
defaultpid:  ipv4:0.0.0.0/0  ipv6:::0/0
pid1:        ipv4:192.0.2.0/25
pid2:        ipv4:192.0.2.0/27
pid3:        ipv4:192.0.3.0/28
pid4:        ipv4:192.0.3.16/28
```

Figure 3: Example Default Network Map

And another simple alternative network map:

```
defaultpid:  ipv4:0.0.0.0/0  ipv6:::0/0
pid1:        ipv4:192.0.2.0/27
pid2:        ipv4:192.0.3.0/27
```

Figure 4: Example Alternative Network Map

10.2. Property Definitions

Beyond "pid", the examples in this section use four additional properties for Internet address domains, "ISP", "ASN", "country" and "state", with the following values:

	ISP	ASN	country	state
ipv4:192.0.2.0/23:	BitsRus	-	us	-
ipv4:192.0.2.0/28:	-	12345	-	NJ
ipv4:192.0.2.16/28:	-	12345	-	CT
ipv4:192.0.2.1:	-	-	-	PA
ipv4:192.0.3.0/28:	-	12346	-	TX
ipv4:192.0.3.16/28:	-	12346	-	MN

Figure 5: Example Property Values for Internet Address Domains

And the examples in this section use the property "region" for the PID domain of the default network map with the following values:

```
                          region
     pid:defaultpid:     -
     pid:pid1:           us-west
     pid:pid2:           us-east
     pid:pid3:           us-south
     pid:pid4:           us-north
```

Figure 6: Example Property Values for Default Network Map's PID Domain

Note that "-" means the value of the property for the entity is "undefined". So the entity would inherit a value for this property by the inheritance rule if possible. For example, the value of the "ISP" property for "ipv4:192.0.2.1" is "BitsRus" because of "ipv4:192.0.2.0/24". But the "region" property for "pid:defaultpid" has no value because no entity from which it can inherit.

Similar to the PID domain of the default network map, the examples in this section use the property "ASN" for the PID domain of the alternative network map with the following values:

```
                          ASN
     pid:defaultpid:       -
     pid:pid1:           12345
     pid:pid2:           12346
```

Figure 7: Example Property Values for Alternative Network Map's PID Domain

10.3. Properties for Abstract Network Elements

Additionally, the examples in this section consider a facilitated entity domain: "ane" (Abstract Network Element). Abstract network elements allow ALTO clients to discover information beyond the end-to-end routing costs. Examples of abstract network elements include:

Forwarding elements: Forwarding elements include optical wires, physical layer links, IP tunnels, etc. Forwarding elements share the common property "maxresbw".

Value-added services: Value-added services include HTTP caches, 5G UPF nodes, mobile edge computing, etc. Value-added services share the common property "persistent-entities", which contains information that points to the entry point of the service. Different value-added services may have specific properties, e.g., an abstract network element of a mobile edge may provide a list of flavors to the client.

	maxresbw	persistent-entities	mec-flavors
ane:L001	100 Mbps		
ane:L002	100 Mbps		
ane:CACHE1		http-proxy:192.0.2.1	
ane:MEC01		mec:192.0.2.1	{gpu:2G, ssd:128G}
ane:MEC02		mec:192.0.2.2	{gpu:1G, ssd:128G}

The "ane" entities are usually not used alone, but associated with other ALTO resources, e.g., cost maps. It means that the ALTO server may not define a property map resource to provide properties of "ane" entities. The property map payload for "ane" entities may be provided in the response of other ALTO resources in some way.

10.4. Information Resource Directory (IRD)

The following IRD defines the relevant resources of the ALTO server. It provides two property maps, one for the "ISP" and "ASN" properties, and another for the "country" and "state" properties. The server could have provided a single property map for all four properties, but did not, presumably because the organization that runs the ALTO server believes any given client is not interested in all four properties.

The server provides two filtered property maps. The first returns all four properties, and the second just returns the "pid" property for the default network map.

The filtered property maps for the "ISP", "ASN", "country" and "state" properties do not depend on the default network map (it does not have a "uses" capability), because the definitions of those properties do not depend on the default network map. The Filtered Property Map for the "pid" property does have a "uses" capability for the default network map, because that defines the values of the "pid" property.

Note that for legacy clients, the ALTO server provides an Endpoint Property Service for the "pid" property for the default network map.

The server also provides a facilitated ALTO resource which accepts the filtered cost map request but returns a multipart message including a cost map and an associated property map for "ane" entities.


```
    "meta" : {
      ...
      "default-alto-network-map" : "default-network-map"
    },
    "resources" : {
```

```
        "default-network-map" : {
          "uri" : "http://alto.example.com/networkmap/default",
          "media-type" : "application/alto-networkmap+json"
        },
        "alt-network-map" : {
          "uri" : "http://alto.example.com/networkmap/alt",
          "media-type" : "application/alto-networkmap+json"
        },
        .... property map resources ....
        "ia-property-map" : {
          "uri" : "http://alto.example.com/propmap/full/inet-ia",
          "media-type" : "application/alto-propmap+json",
          "uses": [ "default-network-map", "alt-network-map" ],
          "capabilities" : {
            "mappings": {
              "ipv4": [ ".ISP", ".ASN" ],
              "ipv6": [ ".ISP", ".ASN" ]
            }
          }
        },
        "iacs-property-map" : {
          "uri" : "http://alto.example.com/propmap/lookup/inet-iacs",
          "media-type" : "application/alto-propmap+json",
          "accepts": "application/alto-propmapparams+json",
          "uses": [ "default-network-map", "alt-network-map" ],
          "capabilities" : {
            "mappings": {
              "ipv4": [ ".ISP", ".ASN", ".country", ".state" ],
              "ipv6": [ ".ISP", ".ASN", ".country", ".state" ]
            }
          }
        },
        "region-property-map": {
          "uri": "http://alto.example.com/propmap/lookup/region",
          "media-type": "application/alto-propmap+json",
          "accepts": "application/alto-propmapparams+json",
          "uses" : [ "default-network-map", "alt-network-map" ],
          "capabilities": {
            "mappings": {
              "default-network-map.pid": [ ".region" ],
              "alt-network-map.pid": [ ".ASN" ],
            }
          }
        },
        "ip-pid-property-map" : {
          "uri" : "http://alto.example.com/propmap/lookup/pid",
          "media-type" : "application/alto-propmap+json",
          "accepts" : "application/alto-propmapparams+json",
```

```
          "uses" : [ "default-network-map", "alt-network-map" ],
          "capabilities" : {
            "mappings": {
              "ipv4": [ "default-network-map.pid",
                        "alt-network-map.pid" ],
              "ipv6": [ "default-network-map.pid",
                        "alt-network-map.pid" ]
            }
          }
        },
        "legacy-endpoint-property" : {
          "uri" : "http://alto.example.com/legacy/eps-pid",
          "media-type" : "application/alto-endpointprop+json",
          "accepts" : "application/alto-endpointpropparams+json",
          "capabilities" : {
            "properties" : [ "default-network-map.pid",
                             "alt-network-map.pid" ]
          }
        },
        "ane-dc-property-map": {
          "uri" : "http://alto.example.com/propmap/lookup/ane-dc",
          "media-type" : "application/alto-propmap+json",
          "accepts": "application/alto-propmapparams+json",
          "capabilities": {
            "mappings": {
              ".ane" : [ "storage-capacity", "ram", "cpu" ]
            }
          },
        }
      }
```


Figure 8: Example IRD

10.5. Full Property Map Example

The following example uses the properties and IRD defined above in Section 10.4 to retrieve a Property Map for entities with the "ISP" and "ASN" properties.

Note that, to be compact, the response does not include the entity "ipv4:192.0.2.0", because values of all those properties for this entity are inherited from other entities.

Also note that the entities "ipv4:192.0.2.0/28" and "ipv4:192.0.2.16/28" are merged into "ipv4:192.0.2.0/27", because they have the same value of the "ASN" property. The same rule applies to the entities "ipv4:192.0.3.0/28" and "ipv4:192.0.3.0/28". Both of "ipv4:192.0.2.0/27" and "ipv4:192.0.3.0/27" omit the value for the "ISP" property, because it is inherited from "ipv4:192.0.2.0/23".

```
GET /propmap/full/inet-ia 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
```


```
{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.0/23":   {".ISP": "BitsRus"},
    "ipv4:192.0.2.0/27":   {".ASN": "12345"},
    "ipv4:192.0.3.0/27":   {".ASN": "12346"}
  }
}
```


10.6. Filtered Property Map Example #1

The following example uses the filtered property map resource to request the "ISP", "ASN" and "state" properties for several IPv4 addresses.

Note that the value of "state" for "ipv4:192.0.2.0" is the only explicitly defined property; the other values are all derived by the inheritance rules for Internet address entities.

```
POST /propmap/lookup/inet-iacs HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: ###
Content-Type: application/alto-propmapparams+json

{
  "entities" : [ "ipv4:192.0.2.0",
                 "ipv4:192.0.2.1",
                 "ipv4:192.0.2.17" ],
  "properties" : [ ".ISP", ".ASN", ".state" ]
}

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

{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.0":
           {".ISP": "BitsRus", ".ASN": "12345", ".state": "PA"},
    "ipv4:192.0.2.1":
           {".ISP": "BitsRus", ".ASN": "12345", ".state": "NJ"},
    "ipv4:192.0.2.17":
           {".ISP": "BitsRus", ".ASN": "12345", ".state": "CT"}
  }
}
```

10.7. Filtered Property Map Example #2

The following example uses the filtered property map resource to request the "ASN", "country" and "state" properties for several IPv4 prefixes.

Note that the property values for both entities "ipv4:192.0.2.0/26" and "ipv4:192.0.3.0/26" are not explicitly defined. They are inherited from the entity "ipv4:192.0.2.0/23".

Also note that some entities like "ipv4:192.0.2.0/28" and "ipv4:192.0.2.16/28" in the response are not listed in the request explicitly. The response includes them because they are refinements of the requested entities and have different values for the requested properties.

The entity "ipv4:192.0.4.0/26" is not included in the response, because there are neither entities which it is inherited from, nor entities inherited from it.

```
POST /propmap/lookup/inet-iacs HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: ###
Content-Type: application/alto-propmapparams+json

{
  "entities" : [ "ipv4:192.0.2.0/26",
                 "ipv4:192.0.3.0/26",
                 "ipv4:192.0.4.0/26" ],
  "properties" : [ ".ASN", ".country", ".state" ]
}

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

{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.0/26":  {".country": "us"},
    "ipv4:192.0.2.0/28":  {".ASN": "12345",
                           ".state": "NJ"},
    "ipv4:192.0.2.16/28": {".ASN": "12345",
                           ".state": "CT"},
    "ipv4:192.0.2.0":     {".state": "PA"},
    "ipv4:192.0.3.0/26":  {".country": "us"},
    "ipv4:192.0.3.0/28":  {".ASN": "12345",
                           ".state": "TX"},
    "ipv4:192.0.3.16/28": {".ASN": "12345",
                           ".state": "MN"}
  }
}
```

10.8. Filtered Property Map Example #3

The following example uses the filtered property map resource to request the "default-network-map.pid" property and the "alt-network-map.pid" property for a set of IPv4 addresses and prefixes.

Note that the entity "ipv4:192.0.3.0/27" is decomposed into two entities "ipv4:192.0.3.0/28" and "ipv4:192.0.3.16/28", as they have different "default-network-map.pid" property values.

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

{
  "entities" : [
                "ipv4:192.0.2.128",
                "ipv4:192.0.2.0/27",
                "ipv4:192.0.3.0/27" ],
  "properties" : [ "default-network-map.pid",
                   "alt-network-map.pid ]
}
```

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

{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.128":   {"default-network-map.pid": "defaultpid",
                           "alt-network-map.pid": "defaultpid"},
    "ipv4:192.0.2.0/27":  {"default-network-map.pid": "pid2",
                           "alt-network-map.pid": "pid1"},
    "ipv4:192.0.3.0/28":  {"default-network-map.pid": "pid3",
                           "alt-network-map.pid": "pid2"},
    "ipv4:192.0.3.16/28": {"default-network-map.pid": "pid4",
                           "alt-network-map.pid": "pid2"}
  }
}
```

10.9. Filtered Property Map Example #4

The following example uses the filtered property map resource to request the "region" property for several PIDs defined in "default-network-map". The value of the "region" property for each PID is not defined by "default-network-map", but the reason why the PID is defined by the network operator.

```
POST /propmap/lookup/region HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: ###
Content-Type: application/alto-propmapparams+json

{
  "entities" : ["default-network-map.pid:pid1",
                "default-network-map.pid:pid2"],
  "properties" : [ ".region" ]
}
```

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


```
{
  "meta" : {
    "dependent-vtags" : [
       {"resource-id": "default-network-map",
        "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"}
    ]
  },
  "property-map": {
    "default-network-map.pid:pid1": {
      ".region": "us-west"
    },
    "default-network-map.pid:pid2": {
      ".region": "us-east"
    }
  }
}
```


10.10. Filtered Property Map for ANEs Example #5

The following example uses the filtered property map resource "ane-dc-property-map" to request properties "storage-capacity" and "cpu" on several ANEs defined in this property map.

```
POST /propmap/lookup/ane-dc HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: ###
Content-Type: application/alto-propmapparams+json
```


```
{
  "entities" : [".ane:dc21", ".ane:dc45.srv9", ".ane:dc6.srv-cluster8"]
  "properties" : [ "storage-capacity", "cpu"]
}
```

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

{
  "meta" : {
  },
  "property-map": {
    ".ane:dc21":
      {"storage-capacity" : 40000 Gbytes, "cpu" : 500 Cores},
    ".ane:dc45.srv9":
      {"storage-capacity" : 100 Gbytes, "cpu" : 20 Cores},
    ".ane:dc6.srv-cluster8":
         {"storage-capacity" : 6000 Gbytes, "cpu" : 100 Cores},
  }
}
```

11. Security Considerations

Both Property Map and Filtered Property Map defined in this document fit into the architecture of the ALTO base protocol, and hence the Security Considerations (Section 15 of [RFC7285]) of the base protocol fully apply: authenticity and integrity of ALTO information (i.e., authenticity and integrity of Property Maps), potential undesirable guidance from authenticated ALTO information (e.g., potentially imprecise or even wrong value of a property such as geo-location), confidentiality of ALTO information (e.g., exposure of a potentially sensitive entity property such as geo-location), privacy for ALTO users, and availability of ALTO services should all be considered.

A particular fundamental security consideration when an ALTO server provides a Property Map is to define precisely the policies on who can access what properties for which entities. Security mechanisms such as authentication and confidentiality mechanisms then should be applied to enforce the policy. For example, a policy can be that a property P can be accessed only by its owner (e.g., the customer who is allocated a given IP address). Then, the ALTO server will need to deploy corresponding mechanisms to realize the policy. The policy may allow non-owners to access a coarse-grained value of the property P. In such a case, the ALTO server may provide a different URI to provide the information.

12. IANA Considerations

This document defines additional application/alto-* media types, and extends the ALTO endpoint property registry.

12.1. application/alto-* Media Types

This document registers two additional ALTO media types, listed in Table 1.

Type	Subtype	Specification
application	alto-propmap+json	Section 7.1
application	alto-propmapparams+json	Section 8.3

Table 1: Additional ALTO Media Types.

Type name: application

Subtype name: This document registers multiple subtypes, as listed in Table 1.

Required parameters: n/a

Optional parameters: n/a

Encoding considerations: Encoding considerations are identical to those specified for the "application/json" media type. See [RFC7159].

Security considerations: Security considerations related to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285].

Interoperability considerations: This document specifies formats of conforming messages and the interpretation thereof.

Published specification: This document is the specification for these media types; see Table 1 for the section documenting each media type.

Applications that use this media type: ALTO servers and ALTO clients either stand alone or are embedded within other applications.

Additional information:

Magic number(s): n/a

File extension(s): This document uses the mime type to refer to protocol messages and thus does not require a file extension.

Macintosh file type code(s): n/a

Person & email address to contact for further information: See Authors' Addresses section.

Intended usage: COMMON

Restrictions on usage: n/a

Author: See Authors' Addresses section.

Change controller: Internet Engineering Task Force (mailto:iesg@ietf.org).

12.2. ALTO Entity Domain Type Registry

This document requests IANA to create and maintain the "ALTO Entity Domain Type Registry", listed in Table 2.

Identifier	Entity Identifier Encoding	Hierarchy & Inheritance
ipv4	See Section 6.1.1	See Section 6.1.3
ipv6	See Section 6.1.2	See Section 6.1.3
pid	See Section 6.2	None

Table 2: ALTO Entity Domains.

This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO entity domains. Second, it states the requirements for allocated entity domains.

12.2.1. Consistency Procedure between ALTO Address Type Registry and ALTO Entity Domain Type Registry

One potential issue of introducing the "ALTO Entity Domain Type Registry" is its relationship with the "ALTO Address Types Registry" already defined in Section 14.4 of [RFC7285]. In particular, the entity identifier of a type of an entity domain registered in the "ALTO Entity Domain Type Registry" MAY match an address type defined in "ALTO Address Type Registry". It is necessary to precisely define and guarantee the consistency between "ALTO Address Type Registry" and "ALTO Entity Domain Registry".

We define that the ALTO Entity Domain Type Registry is consistent with ALTO Address Type Registry if two conditions are satisfied:

- o When an address type is already or able to be registered in the ALTO Address Type Registry [RFC7285], the same identifier MUST be used when a corresponding entity domain type is registered in the ALTO Entity Domain Type Registry.

- o If an ALTO entity domain type has the same identifier as an ALTO address type, their addresses encoding MUST be compatible.

To achieve this consistency, the following items MUST be checked before registering a new ALTO entity domain type in a future document:

- o Whether the ALTO Address Type Registry contains an address type that can be used as an entity identifier for the candidate domain identifier. This has been done for the identifiers "ipv4" and "ipv6" in Table 2.

- o Whether the candidate entity identifier of the type of the entity domain is able to be an endpoint address, as defined in Sections 2.1 and 2.2 of [RFC7285].

When a new ALTO entity domain type is registered, the consistency with the ALTO Address Type Registry MUST be ensured by the following procedure:

- o Test: Do corresponding entity identifiers match a known "network" address type?

  * If yes (e.g., cell, MAC or socket addresses):

    + Test: Is such an address type present in the ALTO Address Type Registry?

      - If yes: Set the new ALTO entity domain type identifier to be the found ALTO address type identifier.

      - If no: Define a new ALTO entity domain type identifier and use it to register a new address type in the ALTO Address Type Registry following Section 14.4 of [RFC7285].

    + Use the new ALTO entity domain type identifier to register a new ALTO entity domain type in the ALTO Entity Domain Type Registry following Section 12.2.2 of this document.

  * If no (e.g., pid name, ane name or country code): Proceed with the ALTO Entity Domain Type registration as described in Section 12.2.2.

12.2.2. ALTO Entity Domain Type Registration Process

New ALTO entity domain types are assigned after IETF Review [RFC5226] to ensure that proper documentation regarding the new ALTO entity domain types and their security considerations has been provided. RFCs defining new entity domain types SHOULD indicate how an entity in a registered type of domain is encoded as an EntityID, and, if applicable, the rules defining the entity hierarchy and property inheritance. Updates and deletions of ALTO entity domains follow the same procedure.

Registered ALTO entity domain type identifiers MUST conform to the syntactical requirements specified in Section 5.1.2. Identifiers are to be recorded and displayed as strings.

Requests to the IANA to add a new value to the registry MUST include the following information:

- o Identifier: The name of the desired ALTO entity domain type.

- o Entity Identifier Encoding: The procedure for encoding the identifier of an entity of the registered type as an EntityID (see Section 5.1.3). If corresponding entity identifiers of an entity domain match a known "network" address type, the Entity Identifier Encoding of this domain identifier MUST include both Address Encoding and Prefix Encoding of the same identifier registered in the ALTO Address Type Registry [RFC7285]. For the purpose of defining properties, an individual entity identifier and the corresponding full-length prefix MUST be considered aliases for the same entity.

- o Hierarchy: If the entities form a hierarchy, the procedure for determining that hierarchy.

- o Inheritance: If entities can inherit property values from other entities, the procedure for determining that inheritance.

- o Media type of defining information resource: some entity domain types allow their entity domain type name to be combined with an information resource name to define a resource-specific entity domain. Such an information resource is called "defining information resource". In this case, the authorized media type of specific information resources MUST be specified in the document defining the entity domain type. When this entity domain type allows combinations with defining resources, this must be indicated and the conditions fully specified in the document. The defining information resource for an entity domain type is the one that:

  * has an entry in the IRD,

  * defines these entities,

  * does not use another information resource that defines these entities,

  * defines and exposes entity identifiers that are all persistent.

  * has a unique media type equal to the one specified in the document defining the entity domain type.

  This information is useful when Servers indicate resource specific entity domains in the property map capabilities. Clients need to know if the combination of information resource type and entity domain type is allowed. See also, Section 4.6 and Section 5.1 for more information.

- Mapping to ALTO Address Type: A boolean value to indicate if the entity domain type can be mapped to the ALTO address type with the same identifier.

- Security Considerations: In some usage scenarios, entity identifiers carried in ALTO Protocol messages may reveal information about an ALTO client or an ALTO service provider. Applications and ALTO service providers using addresses of the registered type should be made aware of how (or if) the addressing scheme relates to private information and network proximity.

This specification requests registration of the identifiers "ipv4", "ipv6" and "pid", as shown in Table 2.

12.3. ALTO Entity Property Type Registry

This document requests IANA to create and maintain the "ALTO Entity Property Type Registry", listed in Table 3.

This registry extends the "ALTO Endpoint Property Type Registry" created by [RFC7285] in that a property is defined on one or more entity domains, rather than just on the IPv4 and IPv6 Internet address domains. entry in this registry is an ALTO entity property type defined in Section 5.2.1. Thus, registered ALTO entity property type identifier MUST conform to the syntactical requirements specified in that section.

Identifier	Intended Semantics
pid	See Section 7.1.1 of [RFC7285]

Table 3: ALTO Entity Property Types.

Requests to the IANA to add a new value to the registry MUST include the following information:

- o Identifier: The unique id for the desired ALTO entity property type. The format MUST be as defined in Section 5.2.1 of this document. It includes the information of the applied ALTO entity domain and the property name.

- o Intended Semantics: ALTO entity properties carry with them semantics to guide their usage by ALTO clients. Hence, a document defining a new type SHOULD provide guidance to both ALTO service providers and applications utilizing ALTO clients as to how values of the registered ALTO entity property should be interpreted.

- o Security Considerations: ALTO entity properties expose information to ALTO clients. ALTO service providers should be made aware of the security ramifications related to the exposure of an entity property.

In security considerations, the request should also discuss the sensitivity of the information, and why such sensitive information is required for ALTO-based operations. Regarding this discussion, the request SHOULD follow the recommendations of Section 14.3. ALTO Endpoint Property Type Registry in [RFC7285].

This document requests registration of the identifier "pid", listed in Table 3. Semantics for this property are documented in Section (TODO: add ref) and security considerations are documented in Section TODO:ref.

13. Acknowledgments

The authors would like to thank discussions with Kai Gao, Qiao Xiang, Shawn Lin, Xin Wang, Danny Perez, and Vijay Gurbani. The authors thank Dawn Chen (Tongji University), and Shenshen Chen (Tongji/Yale University) for their contributions to earlier drafts.

14. References

14.1. 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.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <https://www.rfc-editor.org/info/rfc4632>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/info/rfc5952>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <https://www.rfc-editor.org/info/rfc7159>.

[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>.

[RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <https://www.rfc-editor.org/info/rfc7921>.

[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>.

14.2. Informative References

[draft-ietf-alto-cdni-request-routing-alto] Roome, J., Yang, Y., Ma, K., Peterson, J., and J. Zhang, "Content Delivery Network Interconnection (CDNI) Request Routing: CDNI Footprint and Capabilities Advertisement using ALTO (work in progress)", 2020.

[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-09 (work in progress), November 2019, <http://www.ietf.org/internet-drafts/draft-ietf-alto-path-vector-09.txt>.

Appendix A. Scope of Property Map

Using entity domains to organize entities, an ALTO property map resource can be regarded as given sets of properties for given entity domains. If we ignore the resource-agnostic entity domains, we can regard an ALTO property map resource as a set of (ri, di) => (ro, po) mappings, where (ri, di) means a resource-specific entity domain of type di defined by the information resource ri, and (ro, po) means a resource-specific entity property po defined by the information resource ro.

For each (ri, di) => (ro, po) mapping, the scope of an ALTO property map resource must be one of the cases in the following diagram:

	domain.resource (ri) = r	domain.resource (ri) = this
prop.resource (ro) = r	Export	Non-exist
prop.resource (ro) = this	Extend	Define

where "this" represents the resulting property map resource, and "r" represents an existing ALTO information resource other the resulting property map resource.

- o ri = ro = r ("export" mode): the property map resource just transforms the property mapping di => po defined by r into the unified representation format and exports it. For example: r = "netmap1", di = "ipv4", po = "pid". The property map resource exports the "ipv4 => pid" mapping defined by "netmap1".

- o ri = r, ro = this ("extend" mode): the property map extends properties of entities in the entity domain (r, di) and defines a new property po on them. For example: the property map resource ("this") defines a "geolocation" property on domain "netmap1.pid".

- o ri = ro = this ("define" mode): the property map defines a new intrinsic entity domain and defines property po for each entity in this domain. For example: the property map resource ("this") defines a new entity domain "asn" and defines a property "ipprefixes" on this domain.

- o ri = this, ro = r: in the scope of a property map resource, it does not make sense that another existing ALTO information resource defines a property for this property map resource.

A.1. Example Property Map

The following figure shows an example property map called Property Map 1, which depends on two network maps and provides three sets of mappings by

- o exporting a mapping from ipv4 entities to PIDs defined by two different network maps,

- o extending geo-location properties to ipv4 entities defined by Network Map 1,

- o and defining a new mapping from ASNs to traffic load properties.

```
                                                           (Define)
       +----------+                                     +-------------+
     ->| Property |<---------------------------+--------| asn  | load |
    /  |   Map 1  |                            |        |-------------|
   /   +----------+                            |        | 1234 | 95%  |
  |         ^                                  |        | 5678 | 70%  |
  |         |                                   \       +-------------+
  |         |          (Export)                  \       (Extend)
  |    +---------+    +------------------------+  \     +--------------+
  |    | Network |----| ipv4           | pid    |   -----| geo-location |
  |    |  Map 1  |    |------------------------|        +--------------+
  |    +---------+    | 192.168.0.0/24 | pid1   | - - -> | New York     |
  |                   | 192.168.1.0/24 | pid2   | - - -> | Shanghai     |
  |                   +------------------------+        +--------------+
  |                    (Export)
   \   +---------+    +------------------------+
    ---| Network |----| ipv4           | pid    |
       |  Map 2  |    |------------------------|
       +---------+    | 192.168.0.0/24 | Paris  |
                      | ...            | ...    |
                      +------------------------+
```

More detailed examples are shown in Section 10.

Authors' Addresses

Wendy Roome
Nokia Bell Labs (Retired)
124 Burlington Rd
Murray Hill, NJ 07974
USA

Phone: +1-908-464-6975
Email: wendy@wdroome.com

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
NOZAY 91460
FRANCE

Email: Sabine.Randriamasy@nokia-bell-labs.com

Y. Richard Yang
Yale University
51 Prospect Street
New Haven, CT 06511
USA

Phone: +1-203-432-6400
Email: yry@cs.yale.edu

Jingxuan Jensen Zhang
Tongji University
4800 Caoan Road
Shanghai 201804
China

Email: jingxuan.n.zhang@gmail.com

Kai Gao
Sichuan University
Chengdu 610000
China

Email: kaigao@scu.edu.cn