

CDNI
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

J. Seedorf
HFT Stuttgart - Univ. of Applied Sciences
Y. Yang
Tongji/Yale
K. Ma
Ericsson
J. Peterson
Neustar
X. Lin
Tongji
July 8, 2019

Content Delivery Network Interconnection (CDNI) Request Routing: CDNI
Footprint and Capabilities Advertisement using ALTO
draft-ietf-alto-cdni-request-routing-alto-06

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 2. Background | 4 |
| 2.1. Semantics of FCI Advertisement | 5 |
| 2.2. ALTO Background and Benefits | 6 |
| 3. CDNI FCI Map | 8 |
| 3.1. Media Type | 8 |
| 3.2. HTTP Method | 8 |
| 3.3. Accept Input Parameters | 9 |
| 3.4. Capabilities | 9 |
| 3.5. Uses | 9 |
| 3.6. Response | 9 |
| 3.7. Examples | 11 |
| 3.7.1. IRD Example | 11 |
| 3.7.2. Basic Example | 14 |
| 3.7.3. Incremental Updates Example | 15 |
| 4. CDNI FCI Map using ALTO Network Map | 17 |
| 4.1. Network Map Footprint Type: altonetworkmap | 17 |
| 4.2. Examples | 17 |
| 4.2.1. IRD Example | 17 |
| 4.2.2. ALTO Network Map for CDNI FCI Footprints Example | 17 |
| 4.2.3. ALTO PID Footprints in CDNI FCI | 18 |
| 4.2.4. Incremental Updates Example | 19 |
| 5. Filtered CDNI FCI using Capabilities | 21 |
| 5.1. Media Type | 21 |
| 5.2. HTTP Method | 21 |
| 5.3. Accept Input Parameters | 21 |
| 5.4. Capabilities | 22 |
| 5.5. Uses | 22 |
| 5.6. Response | 22 |
| 5.7. Examples | 23 |
| 5.7.1. IRD Example | 23 |

| | | |
|--------|---|----|
| 5.7.2. | Basic Example | 23 |
| 5.7.3. | Incremental Updates Example | 25 |
| 6. | Query Footprint Properties using ALTO Unified Property Service | 26 |
| 6.1. | Representing Footprint Objects as Unified Property Map Entities | 26 |
| 6.1.1. | ASN Domain | 27 |
| 6.1.2. | COUNTRYCODE Domain | 27 |
| 6.2. | Examples | 28 |
| 6.2.1. | IRD Example | 28 |
| 6.2.2. | Property Map Example | 28 |
| 6.2.3. | Filtered Property Map Example | 30 |
| 6.2.4. | Incremental Updates Example | 31 |
| 7. | Design Decisions and Discussions | 33 |
| 7.1. | Table versus Map | 33 |
| 7.2. | Filter-based Query versus Test-based Query | 33 |
| 8. | IANA Considerations | 34 |
| 8.1. | CDNI Metadata Footprint Type Registry | 34 |
| 8.2. | ALTO Entity Domain Registry | 34 |
| 8.3. | ALTO CDNI FCI Property Type Registry | 34 |
| 9. | Security Considerations | 35 |
| 10. | Acknowledgments | 36 |
| 11. | References | 36 |
| 11.1. | Normative References | 36 |
| 11.2. | Informative References | 37 |
| | Authors' Addresses | 37 |

1. Introduction

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

Among the four interfaces, the CDNI Request Routing Interface provides key functions, as specified in [RFC6707]: "The CDNI Request Routing interface enables a Request Routing function in an Upstream CDN to query a Request Routing function in a Downstream CDN to determine if the Downstream CDN is able (and willing) to accept the delegated Content Request. It also allows the Downstream CDN to control what should be returned to the User Agent in the redirection message by the upstream Request Routing function." On a high level, the scope of the CDNI Request Routing Interface, therefore, contains two main tasks: (1) determining if the downstream CDN (dCDN) is willing to accept a delegated content request; (2) redirecting the

content request coming from an upstream CDN (uCDN) to the proper entry point or entity in the downstream CDN.

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

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

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

The rest of this document is organized as follows. Section 2 provides non-normative background on both CDNI FCI and ALTO. Section 3 introduces the most basic service, called CDNI FCI Map, to realize CDNI FCI using ALTO. Section 4 demonstrates a key benefit of using ALTO: the ability to integrate CDNI FCI with ALTO network maps. Such integration provides a new granularity to describe footprints. Section 5 builds on filtered ALTO maps to introduce filtered CDNI FCI maps using capabilities so that a uCDN can get footprints with given capabilities instead of getting the full map which can be huge. Section 6 further shows a benefit of using ALTO: the ability to query footprint properties using ALTO unified properties. In this way, a uCDN can effectively fetch capabilities of some footprints in which it is interested. IANA and security considerations are discussed in Section 8 and Section 9 respectively.

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

2. Background

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

2.1. Semantics of FCI Advertisement

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

- o Footprint and capabilities are tied together and cannot be interpreted independently from each other. Hence, capabilities must be expressed on a per footprint basis. [RFC8008] integrates footprint and capabilities with an approach of "capabilities with footprint restrictions".
- o Given that a large part of Footprint and Capabilities Advertisement will actually happen in contractual agreements, the semantics of CDNI Footprint and Capabilities advertisement refers to answering the following question: what exactly still needs to be advertised by the CDNI FCI? For instance, updates about temporal failures of part of a footprint can be useful information to convey via the CDNI request routing interface. Such information would provide updates on information previously agreed in contracts between the participating CDNs. In other words, the CDNI FCI is a means for a dCDN to provide changes/updates regarding a footprint and/or capabilities that it has prior agreed to serve in a contract with a uCDN. Hence, server push and incremental encoding will be necessary techniques.
- o Multiple types of footprints (ipv4cidr, ipv6cidr, asn and countrycode) are defined in [RFC8006].
- o A "Set of IP-prefixes" can contain both full IP addresses (i.e., a /32 for IPv4 and a /128 for IPv6) and IP prefixes with an arbitrary prefix length. There must also be support for multiple IP address versions, i.e., IPv4 and IPv6, in such a footprint.
- o For all of these mandatory-to-implement footprint types, footprints can be viewed as constraints for delegating requests to a dCDN: A dCDN footprint advertisement tells the uCDN the limitations for delegating a request to the dCDN. For IP prefixes or ASN(s), the footprint signals to the uCDN that it should consider the dCDN a candidate only if the IP address of the request routing source falls within the prefix set (or ASN, respectively). The CDNI specifications do not define how a given uCDN determines what address ranges are in a particular ASN. Similarly, for country codes, a uCDN should only consider the dCDN

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

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

2.2. ALTO Background and Benefits

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

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

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

- 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. For CDNI, this is exactly the case: a uCDN wants to improve application layer CDN request routing by using dedicated information (provided by a dCDN) that the uCDN could not easily obtain otherwise. ALTO can help a uCDN to select a proper dCDN by first providing dCDNs' capabilities as well as footprints (see

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

- o The semantics of an ALTO network map is an exact match for the needed information to convey a footprint by a downstream CDN, in particular if such a footprint is being expressed by IP-prefix ranges. Please see Section 4.
- o Security: Identifications between uCDNs and dCDNs are extremely important. ALTO maps can be signed and hence provide inherent integrity protection. Please see Section 9.
- 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 has undergone extensive revisions in order to provide sophisticated error-handling, in particular regarding unexpected cases. A CDNI FCI interface based on ALTO would inherit this thought-through and mature error-handling. Please see Section 5.
- o Filtered map service: The ALTO map filtering service would allow a uCDN to query only for parts of an ALTO map. For example, filtered unified property map service can enable a uCDN to query properties of a part of footprints in an effective way (see Section 6).
- o Server-initiated Notifications and Incremental Updates: When the footprint or the capabilities of a downstream CDN change (i.e., unexpectedly from the perspective of an upstream CDN), server-initiated notifications would enable a dCDN to directly inform an upstream CDN about such changes. Consider the case where - due to failure - part of the footprint of the dCDN is not functioning, i.e., the CDN cannot serve content to such clients with reasonable QoS. Without server-initiated notifications, the uCDN might still use a very recent network and cost map from dCDN, and therefore redirect requests to dCDN which it cannot serve. Similarly, the possibility for incremental updates would enable efficient conveyance of the aforementioned (or similar) status changes by the dCDN to the uCDN. The newest design of ALTO supports server pushed incremental updates [I-D.ietf-alto-incr-update-sse].
- 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 an upstream CDN. This would enable a uCDN to determine if a given dCDN actually has the capabilities for a given request with respect to the type of content requested.

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

3. CDNI FCI Map

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

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

3.1. Media Type

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

3.2. HTTP Method

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

3.3. Accept Input Parameters

None.

3.4. Capabilities

None.

3.5. Uses

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

3.6. Response

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

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

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

```
object {  
  CDNIFCIData cdni-fci-map;  
} InfoResourceCDNIFCI : ResponseEntityBase;  
  
object {  
  BaseAdvertisementObject capabilities<1..*>;  
} CDNIFCIData;
```

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

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

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

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

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

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

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

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

EXAMPLE 1

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

```

```

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

```

EXAMPLE 2

```

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

```

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

3.7. Examples

3.7.1. IRD Example

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

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

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

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

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

```

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

```

3.7.2. Basic Example

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

```

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

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

```

```

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

```

3.7.3. Incremental Updates Example

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

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

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

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

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

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

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

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

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



```
data: }

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

4. CDNI FCI Map using ALTO Network Map

4.1. Network Map Footprint Type: altonetworkmap

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

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

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

4.2. Examples

4.2.1. IRD Example

We use the same IRD example given in Section 3.7.1.

4.2.2. ALTO Network Map for CDNI FCI Footprints Example

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

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

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

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

4.2.3. ALTO PID Footprints in CDNI FCI

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

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

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

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

4.2.4. Incremental Updates Example

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

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

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

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

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

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

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

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

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

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

5. Filtered CDNI FCI using Capabilities

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

5.1. Media Type

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

5.2. HTTP Method

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

5.3. Accept Input Parameters

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

```
object {  
  JSONString capability-type;  
  JSONValue capability-value;  
} CDNIFCICapability;  
  
object {  
  [CDNIFCICapability cdni-fci-capabilities<0..*>;]  
} ReqFilteredCDNIFCI;
```

with fields:

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

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

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

5.4. Capabilities

None.

5.5. Uses

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

5.6. Response

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

Specifically, a filtered CDNI FCI request is invalid if:

- 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 FCI capability.

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

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

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

5.7. Examples

5.7.1. IRD Example

We use the same IRD example by Section 3.7.1.

5.7.2. Basic Example

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

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

{
  "cdni-fci-capabilities": [
    {
      "capability-type": "FCI.DeliveryProtocol",
```

```
        "capability-value": {
          "delivery-protocols": [
            "http/1.1"
          ]
        }
      ]
    }
  ]
}

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


5.7.3. Incremental Updates Example

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

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

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

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

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

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

event: application/merge-patch+json,my-fci-stream
data: {
data:   "meta": {
data:     "vtag": {
data:       "tag": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
data:     }
}
```

```

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

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

```

6. Query Footprint Properties using ALTO Unified Property Service

Above sections describe how a uCDN can get the whole capabilities and footprints from dCDNs and how a uCDN can get the footprints of given capabilities. But there is another important case which is how a uCDN can get properties (i.e., capabilities) of given footprints.

The most natural way to solve this problem is to use ALTO unified property map defined in [I-D.ietf-alto-unified-props-new] since footprints can be easily presented as groups of entities and Filtered Property Maps are already well-defined. In this section, we describe how ALTO clients look up properties for individual footprints. We firstly describe how to represent footprint objects as unified property map entities, and then we provide examples of the full property map, the filtered property map and the incremental updates.

6.1. Representing Footprint Objects as Unified Property Map Entities

A footprint object has two properties: footprint-type and footprint-value. A footprint-value is an array of footprint values conforming to the specification associated with the registered footprint type

("ipv4cidr", "ipv6cidr", "asn", and "countrycode"). Since each unified property map entity has a unique address and each pair of footprint-type and a footprint value determines a group of unique addresses, a footprint object can be represented as a set of entities according to their different footprint-type and footprint values. However, [I-D.ietf-alto-unified-props-new] only defines IPv4 Domain and IPv6 Domain which represent footprint-type "ipv4cidr" and "ipv6cidr" respectively. To represent footprint-type "asn" and "countrycode", this document registers two new domains in Section 8.

Here gives an example of representing a footprint object as a set of unified property map entities.

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

6.1.1. ASN Domain

This document specifies a new entity domain type in addition to the ones in [I-D.ietf-alto-unified-props-new]. ASN is the abbreviation of Autonomous System Number.

6.1.1.1. Entity Domain Type

asn

6.1.1.2. Domain-Specific Entity Addresses

The entity address of asn domain is encoded as a string consisting of the characters "as" (in lowercase) followed by the ASN [RFC6793].

6.1.1.3. Hierarchy and Inheritance

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

6.1.2. COUNTRYCODE Domain

This document specifies a new entity domain type in addition to the ones in [I-D.ietf-alto-unified-props-new].

6.1.2.1. Entity Domain Type

countrycode

6.1.2.2. Domain-Specific Entity Addresses

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

6.1.2.3. Hierarchy and Inheritance

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

6.2. Examples

6.2.1. IRD Example

We use the same IRD example given by Section 3.7.1.

6.2.2. Property Map Example

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

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

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

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

6.2.3. Filtered Property Map Example

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

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

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

```

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

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

```

6.2.4. Incremental Updates Example

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

```

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

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

```

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

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

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

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

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

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



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

7. Design Decisions and Discussions

7.1. Table versus Map

A major design decision is if the Map service is suitable to provide the CDNI FCI. Current ALTO protocol uses Map service to provide network information, such as Network Maps, Cost Maps and Property Maps. Their common idea is to use Map-like data structure to represent information. It is different from the data structure of the CDNI FCI designed in [RFC8008], which suggests to use a set of BaseAdvertisementObjects to represent the CDNI FCI information, which actually is Table-like data structure. Both Table and Map can be represented as a set of data entries. But the difference of them is whether there is a primary key to index each data entry.

The main advantage of Map-like data design is to simplify the filter-based query. According to the discussion in [RFC8008] about benefits and concerns of advertisement-based design and query-based design, filter-based query can make the CDNI FCI scalable when the dCDN has thousands or tens of thousands of FCI objects. To transfer Table-like data to Map-like data, introducing the primary key is necessary. This document already defines two different solution to introduce the primary key: (1) set unique identifiers for CDNI capability objects; (2) set unique identifiers for CDNI footprint objects.

But the major concern of the Map-like data design is the redundancy. In Map-like data design, whatever we choose CDNI capability objects or footprint objects as the key, each data entry can only represent the 1-N relation. But there are lots of CDNI FCI objects have the N-N relation.

7.2. Filter-based Query versus Test-based Query

Another design decision is the query approach. ALTO is a query-based protocol. So using ALTO, uCDN should send a query request to the dCDN to pull the CDNI FCI proactively. To make the query efficiently instead of pulling the whole FCI data base every time, query approach design is very important.

This document only defines the filter-based query. A uCDN can specify a set of FCI capability objects or footprint objects to only

query the information including them. But there are two limitations: (1) uCDN cannot filter both of them simultaneously; (2) cannot specify complex filters.

One example is that uCDN wants to filter all CDNI FCI objects whose capabilities are in range C1 and footprints are in range F1, or capabilities are in range C2 and footprints are in range F2.

8. IANA Considerations

8.1. CDNI Metadata Footprint Type Registry

| Footprint Type | Description | Specification |
|----------------|---------------------|---------------|
| altonetworkmap | A list of PID-names | RFCthis |

Table 1: CDNI Metadata Footprint Type

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

8.2. ALTO Entity Domain Registry

As proposed in Section 9.2 of [I-D.ietf-alto-unified-props-new], "ALTO Entity Domain Registry" is requested. Besides, two new domains are to be registered, listed in Table 2.

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

Table 2: ALTO Entity Domain

8.3. ALTO CDNI FCI Property Type Registry

The "ALTO CDNI FCI Property Type Registry" is required by the ALTO Entity Domain "asn", "countrycode", "pid", "ipv4" and "ipv6", listed in Table 3.

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

Table 3: ALTO CDNI FCI Property Type

9. Security Considerations

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

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

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

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

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

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

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

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

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

10. Acknowledgments

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

Jan Seedorf is partially supported by the GreenICN project (GreenICN: Architecture and Applications of Green Information Centric Networking), a research project supported jointly by the European Commission under its 7th Framework Program (contract no. 608518) and the National Institute of Information and Communications Technology (NICT) in Japan (contract no. 167). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the GreenICN project, the European Commission, or NICT.

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>>.

- [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-16 (work in progress), March 2019.
- [I-D.ietf-alto-path-vector]
Gao, K., Lee, Y., Randriamasy, S., Yang, Y., and J. Zhang, "ALTO Extension: Path Vector Cost Type", draft-ietf-alto-path-vector-05 (work in progress), March 2019.
- [I-D.ietf-alto-unified-props-new]
Roome, W., Randriamasy, S., Yang, Y., and J. Zhang, "Unified Properties for the ALTO Protocol", draft-ietf-alto-unified-props-new-07 (work in progress), March 2019.
- [I-D.jenkins-alto-cdn-use-cases]
Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and S. Previdi, "Use Cases for ALTO within CDNs", draft-jenkins-alto-cdn-use-cases-03 (work in progress), June 2012.

Authors' Addresses

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

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

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

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

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

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

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

Email: jon.peterson@neustar.biz

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

Email: x.shawn.lin@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: November 15, 2019

S. Randriamasy
Nokia Bell Labs
R. Yang
Yale University
Q. Wu
Huawei
L. Deng
China Mobile
N. Schwan
Thales Deutschland
May 14, 2019

ALTO Cost Calendar
draft-ietf-alto-cost-calendar-12

Abstract

This document is an extension to the base Application-Layer Traffic Optimization (ALTO) protocol. It extends the ALTO cost information service so that applications decide not only 'where' to connect, but also 'when'. This is useful for applications that need to perform bulk data transfer and would like to schedule these transfers during an off-peak hour, for example. This extension introduces ALTO Cost Calendar, with which an ALTO Server exposes ALTO cost values in JSON arrays where each value corresponds to a given time interval. The time intervals as well as other Calendar attributes, are specified in the Information Resources Directory and ALTO Server responses.

Requirements Language

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.

Status of This Memo

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

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

working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on November 15, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 2. Overview of ALTO Cost Calendars and terminology | 4 |
| 2.1. Terminology | 4 |
| 2.2. ALTO Cost Calendar overview | 5 |
| 2.3. ALTO Cost Calendar information features | 6 |
| 2.4. ALTO Calendar design characteristics | 6 |
| 2.4.1. ALTO Cost Calendar for all cost modes | 7 |
| 2.4.2. Compatibility with legacy ALTO Clients | 8 |
| 3. ALTO Calendar specification: IRD extensions | 8 |
| 3.1. Calendar attributes in the IRD resources capabilities . . | 9 |
| 3.2. Calendars in a delegate IRD | 10 |
| 3.3. Example IRD with ALTO Cost Calendars | 10 |
| 4. ALTO Calendar specification: Service Information Resources . | 14 |
| 4.1. Calendar extensions for Filtered Cost Maps (FCM) | 14 |
| 4.1.1. Calendar extensions in Filtered Cost Map requests . . | 14 |
| 4.1.2. Calendar extensions in Filtered Cost Map responses . | 15 |
| 4.1.3. Use case and example: FCM with a bandwidth Calendar . | 18 |
| 4.2. Calendar extensions in the Endpoint Cost Service | 20 |
| 4.2.1. Calendar specific input in Endpoint Cost requests . | 20 |
| 4.2.2. Calendar attributes in the Endpoint Cost response . . | 20 |
| 4.2.3. Use case and example: ECS with a routingcost Calendar | 21 |

| | |
|--|----|
| 4.2.4. Use case and example: ECS with a multi-cost Calendar for routingcost and owdelay | 24 |
| 5. IANA Considerations | 26 |
| 6. Security Considerations | 26 |
| 7. Operational Considerations | 27 |
| 8. Acknowledgements | 28 |
| 9. References | 28 |
| 9.1. Normative References | 28 |
| 9.2. Informative References | 29 |
| Authors' Addresses | 30 |

1. Introduction

The base Application-Layer Traffic Optimization (ALTO) protocol specified in [RFC7285] provides guidance to overlay applications that need to select one or several hosts from a set of candidates able to provide a desired resource. This guidance is based on parameters that affect performance and efficiency of the data transmission between the hosts such as the topological distance. The goal of ALTO is to improve the Quality of Experience (QoE) in the application while optimizing resource usage in the underlying network infrastructure.

The ALTO protocol in [RFC7285] specifies a network map which defines groupings of endpoints in provider-defined network regions identified by Provider-defined Identifiers (PIDs). The Cost Map Service, Endpoint Cost Service (ECS) and Endpoint Ranking Service then provide ISP-defined costs and rankings for connections among the specified endpoints and PIDs and thus incentives for application clients to connect to ISP preferred locations, for instance, to reduce their costs. For the reasons outlined in the ALTO problem statement [RFC5693] and requirement AR-14 of [RFC6708], ALTO does not disseminate network metrics that change frequently. In a network, the costs can fluctuate for many reasons having to do with instantaneous traffic load or due to diurnal patterns of traffic demand or planned events such as network maintenance, holidays or highly publicized events. Thus, an ALTO application wishing to use the Cost Map and Endpoint Cost Service at some future time will have to estimate the state of the network at that time, a process that is, at best, fragile and brittle since the application does not have any visibility into the state of the network. The need of such future scheduling of large scale traffic that can be addressed by the ALTO protocol is motivated by Unicorn, a unified resource orchestration framework for multi-domain, geo-distributed data analytics, see [draft-xiang-alto-multidomain-analytics].

In case the ALTO Cost value changes are predictable over a certain period of time and the application does not require immediate data

transfer, it can save time to get the whole set of cost values over this period in one single ALTO response. Using this set to schedule data transfers allows optimizing the network resources usage and QoE. ALTO Clients and Servers can also minimize their workload by reducing and accordingly scheduling their data exchanges.

This document extends [RFC7285] to allow an ALTO server to provide network costs for a given duration of time. A sequence of network costs across a time span for a given pair of network locations is named an "ALTO Cost Calendar". The Filtered Cost Map Service and Endpoint Cost Service are extended to provide Cost Calendars. In addition to this functional ALTO enhancement, we expect to further save network and storage resources by gathering multiple Cost Values for one Cost Type into one single ALTO Server response.

In this draft, an "ALTO Cost Calendar" is specified in terms of information resources capabilities that are applicable to time-sensitive ALTO metrics. An ALTO Cost Calendar exposes ALTO Cost Values in JSON arrays, see [RFC8259], where each value corresponds to a given time interval. The time intervals as well as other Calendar attributes are specified in the Information Resources Directory (IRD) and in the Server response to allow the ALTO Client to interpret the received ALTO values. Last, the extensions for ALTO Calendars are applicable to any Cost Mode and they ensure backwards compatibility with legacy ALTO clients.

In the rest of this document, Section 2 provides the design characteristics. Sections 3 and 4 define the formal specifications for the IRD and the information resources. IANA, security and operational considerations are addressed respectively in sections Section 5, Section 6 and Section 7.

2. Overview of ALTO Cost Calendars and terminology

2.1. Terminology

- o ALTO transaction: A request/response exchange between an ALTO Client and an ALTO Server.
- o Client: When used with a capital "C", this term refers to an ALTO Client.
- o Calendar, Cost Calendar: When used with a capital "C", these terms refer to an ALTO Cost Calendar.
- o Endpoint (EP): An endpoint is defined as in Section 2.1 of [RFC7285]. It can be, for example, a peer, a CDN storage location, a physical server involved in a virtual server-supported

application, a party in a resource-sharing swarm such as a computation grid, or an online multi-party game.

- o Server: When used with a capital "S", this term refers to an ALTO Server.

2.2. ALTO Cost Calendar overview

An ALTO Cost Calendar provided by the ALTO Server provides 2 information items:

- o an array of values for a given metric, where each value corresponds to a time interval, where the value array can sometimes be a cyclic pattern that repeats a certain number of times.
- o attributes describing the time scope of the Calendar such as the size and number of the intervals and the date of the starting point of the Calendar, allowing an ALTO Client to interpret the values properly.

An ALTO Cost Calendar can be used like a "time table" to figure out the best time to schedule data transfers and also to proactively manage application traffic given predictable events such as expected spike in traffic due to crowd gathering (concerts, sports, etc.), traffic-intensive holidays and network maintenance. It may be viewed as a synthetic abstraction of, for example, real measurements gathered over previous periods on which statistics have been computed. However, like for any schedule, unexpected network incidents may require the current ALTO Calendar to be updated and re-sent to the ALTO Clients needing it. To this end, it is RECOMMENDED that ALTO Servers providing ALTO Calendars also provide the "ALTO Incremental Updates Using Server-Sent Events (SSE)" Service that is specified in [draft-ietf-alto-incr-update-sse]. Likewise, ALTO Clients capable of using ALTO Calendars SHOULD also use the SSE Service.

Most likely, the ALTO Cost Calendar would be used for the Endpoint Cost Service, assuming that a limited set of feasible Endpoints for a non-real time application is already identified, that they do not need to be accessed immediately and that their access can be scheduled within a given time period. The Filtered Cost Map Service is also applicable as long as the size of the Map allows it.

2.3. ALTO Cost Calendar information features

The Calendar attributes are provided in the Information Resources Directory (IRD) and in ALTO Server responses. The IRD announces attributes without date values in its information resources capabilities, whereas attributes with time dependent values are provided in the "meta" section of Server responses. The ALTO Cost Calendar attributes provide the following information:

- o attributes to describe the time scope of the Calendar value array:
 - * generic time zone,
 - * applicable time interval size for each Calendar value, defined in seconds, that can cover a wide range of values.
 - * duration of the Calendar: e.g., the number of intervals provided in the Calendar.
- o "calendar-start-date": specifying when the Calendar starts, that is to which date the first value of the Cost Calendar is applicable.
- o "repeated": an optional attribute indicating how many iterations of the provided Calendar will have the same values. The server may use it to allow the client to schedule its next request and thus save its own workload by reducing processing of similar requests.

Attribute "repeated" may take a very high value if a Calendar represents a cyclic value pattern that the Server considers valid for a long period. In this case, the Server will only update the Calendar values once this period has elapsed or if an unexpected event occurs on the network.

2.4. ALTO Calendar design characteristics

The extensions in this document encode requests and responses using JSON [RFC8259].

Formally, the cost entries in an ALTO cost map can be any type of JSON value [RFC8259], (see the DstCosts object in Section 11.2.3.6 of [RFC7285]). However, that section states that an implementation of [RFC7285] SHOULD assume that the cost is a JSON number and fail to parse if it is not, unless the implementation is using an extension that signals a different data type. This document extends the definition of a legacy cost map given in [RFC7285] to allow a cost

entry to be an array of values, one per time interval, instead of just one number.

To realize an ALTO Calendar, this document extends: the IRD, the ALTO requests and responses for Cost Calendars.

This extension is designed to be light and to ensure backwards compatibility with base protocol ALTO Clients and with other extensions. It relies on section 8.3.7 "Parsing of Unknown Fields" of [RFC7285] that writes: "Extensions may include additional fields within JSON objects defined in this document. ALTO implementations MUST ignore unknown fields when processing ALTO messages."

The Calendar-specific capabilities are integrated in the information resources of the IRD and in the "meta" member of ALTO responses to Cost Calendars requests. A Calendar and its capabilities are associated with a given information resource and within this information resource with a given cost type. This design has several advantages:

- o it does not introduce a new mode,
- o it does not introduce new media types,
- o it allows an ALTO Server to offer Calendar capabilities on a cost type, with attributes values adapted to each information resource.

The applicable Calendared information resources are:

- o the Filtered Cost Map,
- o the Endpoint Cost Map.

The ALTO Server can choose in which frequency it provides cost Calendars to ALTO Clients. It may either provide Calendar updates starting at the request date, or carefully schedule its updates so as to take profit from a potential repetition/periodicity of Calendar values.

2.4.1. ALTO Cost Calendar for all cost modes

An ALTO Cost Calendar is well-suited for values encoded in the "numerical" mode. Actually, a Calendar can also represent metrics in other modes considered as compatible with time-varying values. For example, types of Cost values such as JSONBool can also be calendared, as their value may be 'true' or 'false' depending on given time periods or likewise, values represented by strings, such as "medium", "high", "low", "blue", "open".

Note also that a Calendar is suitable as well for time-varying metrics provided in the "ordinal" mode, if these values are time-varying and the ALTO Server provides updates of cost value based preferences.

2.4.2. Compatibility with legacy ALTO Clients

The ALTO protocol extensions for Cost Calendars have been defined so as to ensure that Calendar capable ALTO Servers can provide legacy ALTO Clients with legacy information resources as well. That is a legacy ALTO Client can request resources and receive responses as specified in [RFC7285].

A Calendar-aware ALTO Server MUST implement the base protocol specified in [RFC7285].

As a consequence, when a metric is available as a Calendar array, it also MUST be available as a single value as required by [RFC7285]. The Server, in this case, provides the current value of the metric to either Calendar-aware Clients not interested in future or time-based values, or Clients implementing [RFC7285] only.

For compatibility with legacy ALTO Clients specified in [RFC7285], calendared information resources are not applicable for full cost maps for the following reason: a legacy ALTO client would receive a calendared cost map via an HTTP 'GET' command. As specified in section 8.3.7 of [RFC7285], it will ignore the Calendar Attributes indicated in the "meta" of the responses. Therefore, lacking information on Calendar attributes, it will not be able to correctly interpret and process the values of the received array of Calendar cost values.

Therefore, calendared information resources MUST be requested via the Filtered Cost Map Service or the Endpoint Cost Service, using a POST method.

3. ALTO Calendar specification: IRD extensions

The Calendar attributes in the IRD information resources capabilities carry constant dateless values. A Calendar is associated with an information resource rather than a cost type. For example, a Server can provide a "routingcost" Calendar for the Filtered Cost Map Service at a granularity of one day and a "routingcost" Calendar for the Endpoint Cost Service at a finer granularity but for a limited number of endpoints. An example IRD with Calendar specific features is provided in Section 3.3.

3.1. Calendar attributes in the IRD resources capabilities

A Cost Calendar for a given Cost Type MUST be indicated in the IRD by an object of type CalendarAttributes. A CalendarAttribute object is represented by the "calendar-attributes" member of a resource entry. Each CalendarAttributes object applies to a set of one or more cost types. A Cost Type name MUST appear no more than once in the "calendar-attributes" member of a resource entry; multiple appearances of a Cost Type name in CalendarAttributes object of the "calendar-attributes" member MUST cause the ALTO client to ignore any occurrences of this name beyond the first encountered occurrence.

It is RECOMMENDED for an ALTO Server that the time interval size specified in the IRD is the smallest possible one that it can provide. The Client can aggregate cost values on its own if it needs a larger granularity.

The encoding format for object CalendarAttributes, using JSON [RFC8259], is as follows:

```
CalendarAttributes calendar-attributes <1..*>;
```

```
object{
```

```
  JSONString cost-type-names <1..*>;
```

```
  JSONNumber time-interval-size;
```

```
  JSONNumber number-of-intervals;
```

```
} CalendarAttributes;
```

- o "cost-type-names":

- * An array of one or more elements indicating the cost-type-names in the IRD entry to which the capabilities apply.

- o "time-interval-size":

- * is the duration of an ALTO Calendar time interval in seconds. A "time-interval-size" value contains a JSONNumber. ALTO servers SHOULD use at least IEEE 754 double-precision floating point [IEEE.754.2008] to store this value. Example values are: 300 , 7200, meaning that each Calendar value applies on a time interval that lasts respectively 5 minutes and 2 hours.

- o "number-of-intervals":

- * the integer number of values of the Cost Calendar array, at least equal to 1.

- Attribute "cost-type-names" provides a better readability to the Calendar attributes specified in the IRD and avoids confusion with Calendar attributes of other cost-types.
- Multiplying 'time-interval-size' by 'number-of-intervals' provides the duration of the provided Calendar. For example, an ALTO Server may provide a Calendar for ALTO values changing every 'time-interval-size' equal to 5 minutes. If 'number-of-intervals' has the value 12, then the duration of the provided Calendar is "1 hour".

3.2. Calendars in a delegate IRD

It may be useful to distinguish IRD resources supported by the base ALTO protocol from resources supported by its extensions. To achieve this, one option, is that a "root" ALTO Server implementing base protocol resources delegates "specialized" information resources such as the ones providing Cost Calendars, to another ALTO Server running in a subdomain that is specified with its URI in the "root" ALTO Server. This option is described in Section 9.2.4 "Delegation using IRDs" of [RFC7285].

This document provides an example, where a "root" ALTO Server runs in a domain called "alto.example.com". It delegates the announcement of Calendars capabilities to an ALTO Server running in a subdomain called "custom.alto.example.com". The location of the "delegate Calendar IRD" is assumed to be indicated in the "root" IRD by the resource entry: "custom-calendared-resources".

Another advantage is that some Cost Types for some resources may be more advantageous as Cost Calendars and it makes few sense to get them as a single value. For example, Cost Types with predictable and frequently changing values, calendared in short time intervals such as a minute.

3.3. Example IRD with ALTO Cost Calendars

This section provides an example ALTO Server IRD that supports various cost metrics and cost modes. In particular, since [RFC7285] makes it mandatory, the Server uses metric "routingcost" in the "numerical" mode.

For illustrative purposes, this section introduces 3 other fictitious example metrics and modes that should be understood as examples and should not be used or considered as normative.

The cost type names used in the example IRD are thus as follows:

- o "num-routingcost": refers to metric "routingcost" in the numerical mode as defined in [RFC7285] and registered with IANA.
- o "num-owdelay": refers to fictitious performance metric "owdelay" in the "numerical" mode, to reflect the one-way packet transmission delay on a path. A related performance metric is currently under definition in [draft-ietf-alto-performance-metrics].
- o "num-throughputrating": refers to fictitious metric "throughputrating" in the "numerical" mode, to reflect the provider preference in terms of end to end throughput.
- o "string-servicestatus": refers to fictitious metric "servicestatus" in some example mode "string", to reflect the availability, defined by the provider, of for instance path connectivity.

The example IRD includes 2 particular URIs providing Calendars:

- o "https://custom.alto.example.com/calendar/costmap/filtered": a filtered cost map in which Calendar capabilities are indicated for cost type names: "num-routingcost", "num-throughputrating" and "string-servicestatus",
- o "https://custom.alto.example.com/calendar/endpointcost/lookup": an endpoint cost map in which Calendar capabilities are indicated for cost type names: "num-routingcost", "num-owdelay", "num-throughputrating", "string-servicestatus".

The design of the Calendar capabilities allows that some Calendars on a cost type name are available in several information resources with different Calendar Attributes. This is the case for Calendars on "num-routingcost", "num-throughputrating" and "string-servicestatus", available in both the Filtered Cost map and Endpoint Cost Service, but with different time interval sizes for "num-throughputrating" and "string-servicestatus".

--- Client to Server request for IRD -----

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

--- Server response to Client -----

```
HTTP/1.1 200 OK
Content-Length: 2542
```

Content-Type: application/alto-directory+json

```
{
  "meta" : {
    "default-alto-network-map" : "my-default-network-map",
    "cost-types": {
      "num-routingcost": {
        "cost-mode" : "numerical",
        "cost-metric" : "routingcost"
      },
      "num-owdelay": {
        "cost-mode" : "numerical",
        "cost-metric": "owdelay"
      },
      "num-throughputrating": {
        "cost-mode" : "numerical",
        "cost-metric": "throughputrating"
      },
      "string-servicestatus": {
        "cost-mode" : "string",
        "cost-metric": "servicestatus"
      }
    }
  },
  "resources" : {
    "filtered-cost-map-calendar" : {
      "uri" :
        "https://custom.alto.example.com/calendar/costmap/filtered",
      "media-type" : "application/alto-costmap+json",
      "accepts" : "application/alto-costmapfilter+json",
      "capabilities" : {
        "cost-constraints" : true,
        "cost-type-names" : [ "num-routingcost",
                              "num-throughputrating",
                              "string-servicestatus" ],
        "calendar-attributes" : [
          { "cost-type-names" : [ "num-routingcost",
                                "num-throughputrating" ],
            "time-interval-size" : 7200,
            "number-of-intervals" : 12
          },
          { "cost-type-names" : [ "string-servicestatus" ],
            "time-interval-size" : 1800,
            "number-of-intervals" : 48
          }
        ]
      },
      "uses": [ "my-default-network-map" ]
    }
  }
}
```

```

    },
    "endpoint-cost-calendar-map" : {
      "uri" :
        "https://custom.alto.example.com/calendar/endpointcost/lookup",
      "media-type" : "application/alto-endpointcost+json",
      "accepts" : "application/alto-endpointcostparams+json",
      "capabilities" : {
        "cost-constraints" : true,
        "cost-type-names" : [ "num-routingcost",
                              "num-owdelay",
                              "num-throughputrating",
                              "string-servicestatus" ],
        "calendar-attributes" : [
          { "cost-type-names" : [ "num-routingcost" ],
            "time-interval-size" : 3600,
            "number-of-intervals" : 24
          },
          { "cost-type-names" : [ "num-owdelay" ],
            "time-interval-size" : 300,
            "number-of-intervals" : 12
          },
          { "cost-type-names" : [ "num-throughputrating" ],
            "time-interval-size" : 60,
            "number-of-intervals" : 60
          },
          { "cost-type-names" : [ "string-servicestatus" ],
            "time-interval-size" : 120,
            "number-of-intervals" : 30
          }
        ]
      }
    }
  }
}

```

In this example IRD, for the Filtered Cost Map Service:

- o the Calendar for "num-routingcost" and "num-throughputrating" is an array of 12 values each provided on a time interval lasting 7200 seconds (2 hours).
- o the Calendar for "string-servicestatus": "is an array of 48 values each provided on a time interval lasting 1800 seconds (30 minutes).

For the Endpoint Cost Service:

- o the Calendar for "num-routingcost": is an array of 24 values each provided on a time interval lasting 3600 seconds (1 hour).
- o the Calendar for "owdelay": is an array of 12 values each provided on a time interval lasting 300 seconds (5 minutes).
- o the Calendar for "num-throughputrating": is an array of 60 values each provided on a time interval lasting 60 seconds (1 minute).
- o the Calendar for "string-servicestatus": is an array of 30 values each provided on a time interval lasting 120 seconds (2 minutes).

4. ALTO Calendar specification: Service Information Resources

This section documents the individual information resources defined to provide the calendared information services defined in this document.

The reference time zone for the provided time values is UTC. The option chosen to express the time format is the HTTP header fields format specified in [RFC7231] where, however, timestamps are still displayed with the acronym "GMT" rather than "UTC":

Date: Tue, 15 Nov 2014 08:12:31 GMT

The value of a Calendar time interval size is expressed in seconds.

4.1. Calendar extensions for Filtered Cost Maps (FCM)

A legacy ALTO client requests and gets Filtered Cost Map responses as specified in [RFC7285].

4.1.1. Calendar extensions in Filtered Cost Map requests

The input parameters of a "legacy" request for a filtered cost map, defined by object ReqFilteredCostMap in section 11.3.2 of [RFC7285], are augmented with one additional member.

A Calendar-aware ALTO client requesting a Calendar on a given Cost Type for a filtered cost map resource having Calendar capabilities MUST add the following field to its input parameters:

JSONBoolean calendared<1..*>;

This field is an array of 1 to N boolean values, where N is the number of requested metrics. Each entry corresponds to the requested metric at the same array position. Each boolean value indicates

whether or not the ALTO Server should provide the values for this Cost Type as a Calendar. The array MUST contain exactly N boolean values, otherwise, the Server returns an error.

This field MUST NOT be included if no member "calendar-attributes" is specified in this information resource.

If a value of field 'calendared' is 'true' for a cost type name for which no Calendar attributes have been specified: an ALTO Server, whether it implements the extensions of this document or only implements [RFC7285], MUST ignore it and return a response with a single cost value as specified in [RFC7285].

If this field is not present, it MUST be assumed to have only values equal to 'false'.

A Calendar-aware ALTO client that supports requests for only one cost type at a time and wants to request a Calendar MUST provide an array of 1 element:

```
"calendared" : [true];
```

A Calendar-aware ALTO client that supports requests for more than one Cost Types at a time, as specified in [RFC8189] MUST provide an array of N values set to 'true' or 'false', depending whether it wants the applicable Cost Type values as a single or calendared value.

4.1.2. Calendar extensions in Filtered Cost Map responses

In a calendared ALTO Filtered Cost Map, a cost value between a source and a destination is a JSON array of JSON values. An ALTO Calendar values array has a number of values equal to the value of member "number-of-intervals" of the Calendar attributes that are indicated in the IRD. These attributes will be conveyed as metadata in the Filtered Cost Map response. Each element of the array is valid for the time-interval that matches its array position.

The FCM response conveys metadata among which:

- o some are not specific to Calendars and ensure compatibility with [RFC7285] and [RFC8189]
- o some are specific to Calendars.

The non Calendar specific "meta" fields of a calendared Filtered Cost Map response MUST include at least:

- o if the ALTO Client requests cost values for one Cost Type at a time only: the "meta" fields specified in [RFC7285] for these information service responses:
 - * "dependent-vtags ",
 - * "cost-type" field.
- o if the ALTO Client implements the Multi-Cost ALTO extension specified in [RFC8189] and requests cost values for several Cost Types at a time: the "meta" fields specified in [RFC8189] for these information service responses:
 - * "dependent-vtags ",
 - * "cost-type" field with value set to '{}', for backwards compatibility with [RFC7285].
 - * "multi-cost-types" field.

If the client request does not provide member "calendared" or if it provides it with a value equal to 'false', for all the requested Cost Types, then the ALTO Server response is exactly as specified in [RFC7285] and [RFC8189].

If the value of member "calendared" is equal to 'false' for a given requested Cost Type, the ALTO Server MUST return, for this Cost Type, a single cost value as specified in [RFC7285].

If the value of member "calendared" is equal to 'true' for a given requested Cost Type, the ALTO Server returns, for this Cost Type, a cost value Calendar as specified above in this section. In addition to the above cited non Calendar specific "meta" members, the Server MUST provide a Calendar specific metadata field.

The Calendar specific "meta" field that a calendared Filtered Cost Map response MUST include is a member called "calendar-response-attributes", that describes properties of the Calendar and where:

- o member "calendar-response-attributes" is an array of one or more objects of type "CalendarResponseAttributes".
- o each "CalendarResponseAttributes" object in the array is specified for one or more Cost Types for which the value of member "calendared" is equal to 'true' and for which a Calendar is provided for the requested information resource.

- o the "CalendarResponseAttributes" object that applies to a cost type name has a corresponding "CalendarAttributes" object defined for this cost type name in the IRD capabilities of the requested information resource. The members of a "CalendarResponseAttributes" object include all the members of the corresponding "CalendarAttributes" object.

The format of member "CalendarResponseAttributes" is defined as follows:

```
CalendarResponseAttributes calendar-response-attributes <1..*>;
```

```
object{  
  [JSONString cost-type-names <1..*>];  
  JSONString calendar-start-time;  
  JSONNumber time-interval-size;  
  JSONNumber number-of-intervals;  
  [JSONNumber repeated;]  
} CalendarResponseAttributes;
```

Object CalendarResponseAttributes has the following attributes:

- o "cost-type-names": is an array of one or more cost-type-names to which the capabilities apply and for which a Calendar has been requested. The value of this member is a subset of the "cost-type-names" array specified in the corresponding IRD Calendar attributes.
- o "calendar-start-time": indicates the date at which the first value of the Calendar applies. The value provided for the "calendar-start-time" attribute SHOULD NOT be later than the request date.
- o "time-interval-size": as specified in Section 3.1 and with the same value.
- o "number-of-intervals": as specified in Section 3.1 and with the same value.
- o "repeated": is an optional field provided for Calendars. It is an integer N greater or equal to '1' that indicates how many iterations of the Calendar value array starting at the date indicated by "calendar-start-time" have the same values. The number N includes the provided iteration.

For example: suppose the "calendar-start-time" member has value "Mon, 30 Jun 2014 at 00:00:00 GMT", the "time-interval-size" member has value '3600', the "number-of-intervals" member has value '24' and the value of member "repeated" is equal to '4'. This means that the

Calendar values are the same on Monday, Tuesday, Wednesday and Thursday on a period of 24 hours starting at 00:00:00 GMT. The ALTO Client thus may use the same Calendar for the next 4 days starting at "calendar-start-time" and will only need to request a new one for Friday July 4th at 00:00:00 GMT.

Attribute "repeated" may take a very high value if a Calendar represents a cyclic value pattern that the Server considers valid for a long period and hence will only update once this period has elapsed or if an unexpected event occurs on the network. In the latter case, the client will be notified if it uses the "ALTO Incremental Updates Using Server-Sent Events (SSE)" Service, specified in [draft-ietf-alto-incr-update-sse]. See also discussion in Section 7 "Operational Considerations".

4.1.1.3. Use case and example: FCM with a bandwidth Calendar

An example of non-real time information that can be provisioned in a Calendar is the expected path throughput. While the transmission rate can be measured in real time by end systems, the operator of a data center is in the position of formulating preferences for given paths, at given time periods to avoid traffic peaks due to diurnal usage patterns. In this example, we assume that an ALTO Client requests a Calendar of network provider defined throughput ratings, as specified in the IRD, to schedule its bulk data transfers as described in the use cases.

In the example IRD, Calendars for cost type name "num-throughputrating" are available for the information resources: "filtered-cost-calendar-map" and "endpoint-cost-calendar-map". The ALTO Client requests a Calendar for "num-throughputrating" via a POST request for a filtered cost map.

We suppose in the present example that the ALTO Client sends its request on Tuesday July 1st 2014 at 13:15. The Server returns Calendars with arrays of 12 numbers for each source and destination pair. The values for metric "throughputrating", in this example, are assumed to be encoded in 2 digits. For representation brevity, the arrays in the provided example are symbolized by expression "[v1,v2, ... v12]", that is otherwise not valid in JSON. The value of field "Content-Length" in the responses is computed as if "throughputrating" values were encoded in 2 digits. The same type of symbolization is used in the other example Server responses in Section 4.2.3 and Section 4.2.4 of this document.


```
POST /calendar/costmap/filtered HTTP/1.1
Host: alto.example.com
Content-Length: 208
Content-Type: application/alto-costmapfilter+json
Accept: application/alto-costmap+json,application/alto-error+json
```

```
{
  "cost-type" : {"cost-mode" : "numerical",
                 "cost-metric" : "throughputrating"},
  "calendared" : [true],
  "pids" : {
    "srcs" : [ "PID1", "PID2" ],
    "dsts" : [ "PID1", "PID2", "PID3" ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 815
Content-Type: application/alto-costmap+json
```

```
{
  "meta" : {
    "dependent-vtags" : [
      {"resource-id": "my-default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"}
    ],
    "cost-type" : {"cost-mode" : "numerical",
                   "cost-metric" : "throughputrating"},
    "calendar-response-attributes" : [
      {"calendar-start-time" : "Tue, 1 Jul 2014 13:00:00 GMT",
       "time-interval-size" : 7200,
       "number-of-intervals" : 12}
    ]
  },
  "cost-map" : {
    "PID1": { "PID1": [v1,v2, ... v12],
              "PID2": [v1,v2, ... v12],
              "PID3": [v1,v2, ... v12] },
    "PID2": { "PID1": [v1,v2, ... v12],
              "PID2": [v1,v2, ... v12],
              "PID3": [v1,v2, ... v12] }
  }
}
```

4.2. Calendar extensions in the Endpoint Cost Service

This document extends the Endpoint Cost Service, as defined in {11.5.1} of [RFC7285], by adding new input parameters and capabilities, and by returning JSONArrays instead of JSONNumbers as the cost values. The media type {11.5.1.1} and HTTP method {11.5.1.2} are unchanged.

4.2.1. Calendar specific input in Endpoint Cost requests

The extensions to the requests for calendared Endpoint Cost Maps are the same as for the Filtered Cost Map Service, specified in section Section 4.1.1 of this draft.

The ReqEndpointCostMap object for a calendared ECM request will have the following format:

```
object {  
  [CostType cost-type;]  
  [CostType multi-cost-types<1..*>;]  
  [JSONBoolean calendared<1..*>;]  
  EndpointFilter endpoints;  
} ReqEndpointCostMap;  
  
object {  
  [TypedEndpointAddr srcs<0..*>;]  
  [TypedEndpointAddr dsts<0..*>;]  
} EndpointFilter;
```

4.2.2. Calendar attributes in the Endpoint Cost response

The "meta" field of a calendared Endpoint Cost response MUST include at least:

- o if the ALTO Client supports cost values for one Cost Type at a time only: the "meta" fields specified in {11.5.1.6} of [RFC7285] for the Endpoint Cost response:
 - * "cost-type" field.
- o if the ALTO Client supports cost values for several Cost Types at a time, as specified in [RFC8189] : the "meta" fields specified in [RFC8189] for the the Endpoint Cost response:
 - * "cost-type" field with value set to '{}', for backwards compatibility with [RFC7285].

* "multi-cost-types" field.

If the client request does not provide member "calendared" or if it provides it with a value equal to 'false', for all the requested Cost Types, then the ALTO Server response is exactly as specified in [RFC7285] and [RFC8189].

If the ALTO client provides member "calendared" in the input parameters with a value equal to 'true' for given requested Cost Types, the "meta" member of a calendared Endpoint Cost response MUST include, for these Cost Types, an additional member "calendar-response-attributes", the contents of which obey the same rules as for the Filtered Cost Map Service, specified in Section 4.1.2. The Server response is thus changed as follows, w.r.t [RFC7285] and [RFC8189]:

- o the "meta" member has one additional field "CalendarResponseAttributes", as specified for the Filtered Cost Map Service,
- o the calendared costs are JSONArrays instead of the JSONNumbers format used by legacy ALTO implementations. All arrays have a number of values equal to 'number-of-intervals'.

If the value of member "calendared" is equal to 'false' for a given requested Cost Type, the ALTO Server MUST return, for this Cost Type, a single cost value as specified in [RFC7285].

4.2.3. Use case and example: ECS with a routingcost Calendar

Let us assume an Application Client is located in an end system with limited resources and having access to the network that is either intermittent or provides an acceptable quality in limited but predictable time periods. Therefore, it needs to both schedule its resource-greedy networking activities and its ALTO transactions.

The Application Client has the choice to trade content or resources with a set of Endpoints and needs to decide with which one it will connect and at what time. For instance, the Endpoints are spread in different time-zones, or have intermittent access. In this example, the 'routingcost' is assumed to be time-varying, with values provided as ALTO Calendars.

The ALTO Client associated with the Application Client queries an ALTO Calendar on 'routingcost' and will get the Calendar covering the 24 hours time period "containing" the date and time of the ALTO client request.

For Cost Type "num-routingcost", the solicited ALTO Server has defined 3 different daily patterns each represented by a Calendar, to cover the week of Monday June 30th at 00:00 to Sunday July 6th 23:59:

- C1 for Monday, Tuesday, Wednesday, Thursday, (weekdays)
- C2 for Saturday, Sunday, (weekend)
- C3 for Friday (maintenance outage on July 4, 2014 from 02:00:00 GMT to 04:00:00 GMT, or big holiday such as New Year evening).

In the following example, the ALTO Client sends its request on Tuesday July 1st 2014 at 13:15.

The "routingcost" values are assumed to be encoded in 3 digits.

For representation brevity, the arrays in the provided example are symbolized by expression "[v1,v2, ... v24]", that is otherwise not valid in JSON. The value of field "Content-Length" in the responses is computed as if "routingcost" values were encoded in 3 digits.

```
POST /calendar/endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 290
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type" : {"cost-mode" : "numerical",
                 "cost-metric" : "routingcost"},
  "calendared" : [true],
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45",
      "ipv6:2001:db8::10"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 877
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta" : {
    "cost-type" : {"cost-mode" : "numerical",
                   "cost-metric" : "routingcost"},
    "calendar-response-attributes" : [
      {"calendar-start-time" : "Mon, 30 Jun 2014 00:00:00 GMT",
       "time-interval-size" : 3600,
       "number-of-intervals" : 24,
       "repeated": 4
      }
    ]
  },
  "endpoint-cost-map" : {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89" : [v1, v2, ... v24],
      "ipv4:198.51.100.34" : [v1, v2, ... v24],
      "ipv4:203.0.113.45" : [v1, v2, ... v24],
      "ipv6:2001:db8::10" : [v1, v2, ... v24]
    }
  }
}
```

When the Client gets the Calendar for "routingcost", it sees that the "calendar-start-time" is Monday at 00h00 GMT and member "repeated" is equal to '4'. It understands that the provided values are valid until Thursday included and will only need to get a Calendar update on Friday.

4.2.4. Use case and example: ECS with a multi-cost Calendar for routingcost and owdelay

In this example, it is assumed that the ALTO Server implements multi-cost capabilities, as specified in [RFC8189]. That is, an ALTO client can request and receive values for several cost types in one single transaction. An illustrating use case is a path selection done on the basis of 2 metrics: routing cost and owdelay.

As in the previous example, the IRD indicates that the ALTO Server provides "routingcost" Calendars in terms of 24 time intervals of 1 hour (3600 seconds) each.

For metric "owdelay", the IRD indicates that the ALTO Server provides Calendars in terms of 12 time intervals values lasting each 5 minutes (300 seconds).

In the following example transaction, the ALTO Client sends its request on Tuesday July 1st 2014 at 13:15.

This example assumes that the values of metric "owdelay" and "routingcost" are encoded in 3 digits.

For representation brevity, the arrays in the provided example are symbolized by expression "[[r1, r2, ... r24], [o1, o2, ... o12]]", that is otherwise not valid in JSON. The value of field "Content-Length" in the responses is computed as if "routingcost" and "owdelay" values were encoded in 3 digits.

```
POST calendar/endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 373
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type" : {},
  "multi-cost-types" : [
    {"cost-mode" : "numerical", "cost-metric" : "routingcost"},
    {"cost-mode" : "numerical", "cost-metric" : "owdelay"}
  ],
}
```

```

    "calendared" : [true, true],
    "endpoints" : {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv4:203.0.113.45",
        "ipv6:2001:db8::10"
      ]
    }
  }
}

```

HTTP/1.1 200 OK

Content-Length: 1377

Content-Type: application/alto-endpointcost+json

```

{
  "meta" : {
    "multi-cost-types" : [
      { "cost-mode" : "numerical", "cost-metric" : "routingcost" },
      { "cost-mode" : "numerical", "cost-metric" : "owdelay" }
    ],
    "calendar-response-attributes" : [
      { "cost-type-names" : "num-routingcost",
        "calendar-start-time" : "Mon, 30 Jun 2014 00:00:00 GMT",
        "time-interval-size" : 3600,
        "number-of-intervals" : 24,
        "repeated": 4 },
      { "cost-type-names" : "num-owdelay",
        "calendar-start-time" : "Tue, 1 Jul 2014 13:00:00 GMT",
        "time-interval-size" : 300,
        "number-of-intervals" : 12 }
    ],
  },
  "endpoint-cost-map" : {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89" : [[r1, r2, ... r24], [o1, o2, ... o12]],
      "ipv4:198.51.100.34" : [[r1, r2, ... r24], [o1, o2, ... o12]],
      "ipv4:203.0.113.45" : [[r1, r2, ... r24], [o1, o2, ... o12]],
      "ipv6:2001:db8::10" : [[r1, r2, ... r24], [o1, o2, ... o12]]
    }
  }
}

```

When receiving the response, the client sees that the Calendar values for 'routing cost' are repeated for 4 iterations. Therefore, in its

next requests until the routing cost Calendar is expected to change, the client will only need to request a Calendar for "owdelay".

Without the ALTO Calendar extensions, the ALTO client would have no clue on the dynamicity of the metric value change and would spend needless time requesting values at an inappropriate pace. In addition, without the Multi-Cost ALTO capabilities, the ALTO client would duplicate this waste of time as it would need to send one request per cost metric.

5. IANA Considerations

This document does not define any new media types or introduce any new IANA considerations.

6. Security Considerations

As an extension of the base ALTO protocol [RFC7285], this document fits into the architecture of the base protocol, and hence the Security Considerations (Section 15) of the base protocol fully apply when this extension is provided by an ALTO server. For example, the same authenticity and integrity considerations (Section 15.1 of [RFC7285]) still fully apply; the same considerations for the privacy of ALTO users (Section 15.4 of [RFC7285]) also still fully apply.

The calendaring information provided by this extension requires additional considerations on three security considerations discussed in the base protocol: potential undesirable guidance to clients (Section 15.2 of [RFC7285]), confidentiality of ALTO information (Section 15.2 of [RFC7285]), and availability of ALTO (Section 15.5 of [RFC7285]). For example, by providing network information in the future in a Calendar, this extension may improve availability of ALTO, when the ALTO server is unavailable but related information is already provided in the Calendar.

For confidentiality of ALTO information, an operator should be cognizant that this extension may introduce a new risk: an ALTO client may get information for future events that are scheduled through Calendaring. Possessing such information, the client may use it to achieve its goal: (1) initiating connections only at advantageous network costs, leading to unexpected network load; (2) generating massive connections to the network at times where its load is expected to be high.

To mitigate this risk, the operator should address the risk of ALTO information being leaked to malicious clients or third parties. As specified in Section 15.3.2 ("Protection Strategies") of [RFC7285], the ALTO server should authenticate ALTO clients and use the

Transport Layer Security (TLS) protocol so that Man In The Middle (MITM) attacks to intercept an ALTO Calendar are not possible. [RFC7285] ensures the availability of such a solution in its Section 8.3.5. "Authentication and Encryption", which specifies that: "ALTO server implementations as well as ALTO client implementations MUST support the "https" URI scheme of [RFC2818] and Transport Layer Security (TLS) of [RFC5246]".

[RFC8446] specifies TLS 1.3 and writes in its section 1: "While TLS 1.3 is not directly compatible with previous versions, all versions of TLS incorporate a versioning mechanism which allows clients and servers to interoperably negotiate a common version if one is supported by both peers". So ALTO clients and servers MAY use newer versions (e.g., 1.3) of TLS as long as the negotiation process succeeds. To ensure backward compatibility with [RFC7285], it is RECOMMENDED for both Calendar-aware Clients and Servers to both support at least TLS 1.2, until it gets deprecated.

To avoid malicious or erroneous guidance from ALTO information, an ALTO client should be cognizant that using calendaring information can have risks: (1) Calendar values, especially in "repeated" Calendars may be only statistical, and (2) future events may change. Hence, a more robust ALTO client should adapt and extend protection strategies specified in Section 15.2 of the base protocol: it should develop self-check and also ensure information update, to reduce the impact of this risk. To address the risk of unexpected ALTO Values changes that the ALTO Client would be unaware of, it is RECOMMENDED that Servers supporting Calendars also support the "ALTO Incremental Updates Using Server-Sent Events (SSE)" Service, specified in [draft-ietf-alto-incr-update-sse]. Likewise, it is RECOMMENDED that Clients using Calendars also support the SSE Service.

7. Operational Considerations

Conveying ALTO Cost Calendars tends to reduce the on-the-wire data exchange volume compared to multiple single cost ALTO transactions. An application using Calendars has a set of time-dependent values upon which it can plan its connections in advance with no need for the ALTO Client to query information at each time. Additionally, the Calendar response attribute "repeated", when provided, saves additional data exchanges in that it indicates that the ALTO Client does not need to query Calendars during a period indicated by this attribute. Unexpected changes during this period can be handled by using the SSE Service as discussed in Section 6, if the Server and the Client support it.

High-resolution intervals may be needed when values change, sometimes during very small time intervals but in a significant manner. A way

to avoid conveying too many entries is to leverage on the "repeated" feature. A server can smartly set the Calendar start time and number of intervals so as to declare them "repeated" for a large number of periods, until the Calendar values change and are conveyed to requesting Clients.

Clients and Servers supporting ALTO Calendars use [RFC8259]. [RFC7285] encodes its requests and responses using the JSON Data Interchange Format specified in [RFC7159]. In the meantime, [RFC7159] has been obsoleted by [RFC8259], that among others makes UTF-8 mandatory for text encoding to improve interoperability. Therefore, ALTO Clients and Servers implementations using UTF-{16,32} need to be cognizant of the subsequent interoperability risks and it is RECOMMENDED for them to switch to UTF-8 encoding, if they want to interoperate with Calendar-aware Servers and Clients.

8. Acknowledgements

The authors would like to thank Fred Baker, Li Geng, Diego Lopez, He Peng and Haibin Song for fruitful discussions and feedback on earlier draft versions. Dawn Chan, Kai Gao, Vijay Gurbani, Yichen Qian and Jensen Zhang provided substantial review feedback and suggestions to the protocol design.

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>>.
- [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>>.

- [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

- [draft-ietf-alto-incr-update-sse]
"ALTO Incremental Updates Using Server-Sent Events (SSE) (work in progress)", December 2018.
- [draft-ietf-alto-performance-metrics]
"ALTO Performance Cost Metrics (work in progress)", June 2018.
- [draft-xiang-alto-multidomain-analytics]
"Unicorn: Resource Orchestration for Multi-Domain, Geo-Distributed Data Analytics", July 2018.
- [IEEE.754.2008]
"Standard for Binary Floating-Point Arithmetic, IEEE Standard 754", August 2008.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, DOI 10.17487/RFC2818, May 2000, <<https://www.rfc-editor.org/info/rfc2818>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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>>.
- [RFC6708] Kiesel, S., Ed., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", RFC 6708, DOI 10.17487/RFC6708, September 2012, <<https://www.rfc-editor.org/info/rfc6708>>.
- [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>>.

- [RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<https://www.rfc-editor.org/info/rfc7231>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

Authors' Addresses

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
NOZAY 91460
FRANCE

Email: Sabine.Randriamasy@nokia-bell-labs.com

Richard Yang
Yale University
51 Prospect st
New Haven, CT 06520
USA

Email: yry@cs.yale.edu

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

Email: sunseawq@huawei.com

Lingli Deng
China Mobile
China

Email: denglingli@chinamobile.com

Nico Schwan
Thales Deutschland
Lorenzstrasse 10
Stuttgart 70435
Germany

Email: nico.schwan@thalesgroup.com

ALTO
Internet-Draft
Intended status: Experimental
Expires: 21 September 2022

K. Gao
Sichuan University
Y. Lee
Samsung
S. Randriamasy
Nokia Bell Labs
Y.R. Yang
Yale University
J. Zhang
Tongji University
20 March 2022

An ALTO Extension: Path Vector
draft-ietf-alto-path-vector-25

Abstract

This document is an extension to the base Application-Layer Traffic Optimization (ALTO) protocol. It extends the ALTO Cost Map and ALTO Property Map services so that an application can decide which endpoint(s) to connect based on not only numerical/ordinal cost values but also fine-grained abstract information of the paths. This is useful for applications whose performance is impacted by specified components of a network on the end-to-end paths, e.g., they may infer that several paths share common links and prevent traffic bottlenecks by avoiding such paths. This extension introduces a new abstraction called Abstract Network Element (ANE) to represent these components and encodes a network path as a vector of ANEs. Thus, it provides a more complete but still abstract graph representation of the underlying network(s) for informed traffic optimization among endpoints.

Status of This Memo

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

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

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

This Internet-Draft will expire on 21 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 4 |
| 2. Requirements Languages | 6 |
| 3. Terminology | 6 |
| 4. Requirements and Use Cases | 7 |
| 4.1. Design Requirements | 7 |
| 4.2. Sample Use Cases | 10 |
| 4.2.1. Exposing Network Bottlenecks | 11 |
| 4.2.2. Resource Exposure for CDN and Service Edge | 15 |
| 5. Path Vector Extension: Overview | 17 |
| 5.1. Abstract Network Element (ANE) | 18 |
| 5.1.1. ANE Entity Domain | 19 |
| 5.1.2. Ephemeral and Persistent ANEs | 19 |
| 5.1.3. Property Filtering | 20 |
| 5.2. Path Vector Cost Type | 20 |
| 5.3. Multipart Path Vector Response | 21 |
| 5.3.1. Identifying the Media Type of the Root Object | 22 |
| 5.3.2. References to Part Messages | 22 |
| 6. Specification: Basic Data Types | 23 |
| 6.1. ANE Name | 23 |
| 6.2. ANE Entity Domain | 23 |
| 6.2.1. Entity Domain Type | 23 |
| 6.2.2. Domain-Specific Entity Identifier | 23 |
| 6.2.3. Hierarchy and Inheritance | 23 |
| 6.2.4. Media Type of Defining Resource | 23 |
| 6.3. ANE Property Name | 24 |
| 6.4. Initial ANE Property Types | 24 |
| 6.4.1. Maximum Reservable Bandwidth | 24 |
| 6.4.2. Persistent Entity ID | 25 |
| 6.4.3. Examples | 25 |
| 6.5. Path Vector Cost Type | 26 |

| | | |
|-------------|--|----|
| 6.5.1. | Cost Metric: ane-path | 26 |
| 6.5.2. | Cost Mode: array | 27 |
| 6.6. | Part Resource ID and Part Content ID | 27 |
| 7. | Specification: Service Extensions | 27 |
| 7.1. | Notations | 27 |
| 7.2. | Multipart Filtered Cost Map for Path Vector | 28 |
| 7.2.1. | Media Type | 28 |
| 7.2.2. | HTTP Method | 28 |
| 7.2.3. | Accept Input Parameters | 28 |
| 7.2.4. | Capabilities | 29 |
| 7.2.5. | Uses | 30 |
| 7.2.6. | Response | 30 |
| 7.3. | Multipart Endpoint Cost Service for Path Vector | 34 |
| 7.3.1. | Media Type | 34 |
| 7.3.2. | HTTP Method | 34 |
| 7.3.3. | Accept Input Parameters | 34 |
| 7.3.4. | Capabilities | 35 |
| 7.3.5. | Uses | 35 |
| 7.3.6. | Response | 35 |
| 8. | Examples | 39 |
| 8.1. | Sample Setup | 39 |
| 8.2. | Information Resource Directory | 39 |
| 8.3. | Multipart Filtered Cost Map | 42 |
| 8.4. | Multipart Endpoint Cost Service Resource | 43 |
| 8.5. | Incremental Updates | 48 |
| 8.6. | Multi-cost | 50 |
| 9. | Compatibility with Other ALTO Extensions | 52 |
| 9.1. | Compatibility with Legacy ALTO Clients/Servers | 53 |
| 9.2. | Compatibility with Multi-Cost Extension | 53 |
| 9.3. | Compatibility with Incremental Update | 53 |
| 9.4. | Compatibility with Cost Calendar | 53 |
| 10. | General Discussions | 54 |
| 10.1. | Constraint Tests for General Cost Types | 54 |
| 10.2. | General Multi-Resource Query | 54 |
| 11. | Security Considerations | 55 |
| 12. | IANA Considerations | 57 |
| 12.1. | ALTO Cost Metric Registry | 57 |
| 12.2. | ALTO Cost Mode Registry | 58 |
| 12.3. | ALTO Entity Domain Type Registry | 58 |
| 12.4. | ALTO Entity Property Type Registry | 59 |
| 12.4.1. | New ANE Property Type: Maximum Reservable Bandwidth | 59 |
| 12.4.2. | New ANE Property Type: Persistent Entity ID | 60 |
| 13. | References | 60 |
| 13.1. | Normative References | 60 |
| 13.2. | Informative References | 61 |
| Appendix A. | Acknowledgments | 64 |
| Appendix B. | Revision Logs (To be removed before publication) | 64 |

| | | |
|--------------------|-------------------------------------|----|
| B.1. | Changes since -20 | 64 |
| B.2. | Changes since -19 | 65 |
| B.3. | Changes since -18 | 65 |
| B.4. | Changes since -17 | 65 |
| B.5. | Changes since -16 | 65 |
| B.6. | Changes since -15 | 65 |
| B.7. | Changes since -14 | 65 |
| B.8. | Changes since -13 | 66 |
| B.9. | Changes since -12 | 66 |
| B.10. | Changes since -11 | 66 |
| B.11. | Changes since -10 | 66 |
| B.12. | Changes since -09 | 67 |
| B.13. | Changes since -08 | 67 |
| B.14. | Changes Since Version -06 | 67 |
| Authors' Addresses | | 68 |

1. Introduction

Network performance metrics are crucial to assess the Quality of Experience (QoE) of 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 perceived QoE by better orchestrating their traffic to utilize the resources in the underlying network infrastructure.

Existing ALTO Cost Map (Section 11.2.3 of [RFC7285]) and Endpoint Cost Service (Section 11.5 of [RFC7285]) 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 [RFC8896].

While the existing extensions are sufficient for many overlay applications, the QoE of some overlay applications depends not only on the cost information of end-to-end paths, but also on particular components of a network on the paths 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 bottleneck links inside the carrier network as link capacity may impact the rate of data input/output to the job. We refer to such components of a network as Abstract Network Elements (ANE).

Predicting such information can be very complex without the help of ISPs, for example, [BOXOPT] has shown that finding the optimal bandwidth reservation for multiple flows can be NP-hard without further information than whether a reservation succeeds. 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 claimed benefits, ISPs are not likely to expose raw details on their network paths: first for the sake of topology hiding requirement, second because it may increase volume and computation overhead, and last because applications do not necessarily need all the network path details and are likely not able to understand them.

Therefore, it is beneficial for both ISPs and applications if an ALTO server provides ALTO clients with an "abstract network state" that provides the necessary information to applications, while hiding the network complexity and confidential information. An "abstract network state" is a selected set of abstract representations of Abstract Network Elements traversed by the paths between <source, destination> pairs combined with properties of these Abstract Network Elements 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 Abstract Network Elements. Server scalability can also be improved by combining Abstract Network Elements and their properties in a single response.

This document extends [RFC7285] to allow an ALTO server to convey "abstract network state", for paths defined by their <source, destination> pairs. To this end, it introduces a new cost type called "Path Vector" following the cost metric registration specified in [RFC7285] and the updated cost mode registration specified in [I-D.bw-alto-cost-mode]. A Path Vector is an array of identifiers that identifies an Abstract Network Element, which 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 of an ALTO server when providing Path Vector service. 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 involves 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" content type defined in [RFC2387] to return the two maps in a single response.

As a single ISP may not have the knowledge of the full Internet paths between arbitrary endpoints, this document is mainly applicable 1) when there is a single ISP between the requested source and destination PIDs or endpoints, for example, ISP-hosted CDN/edge, tenant interconnection in a single public cloud platform, etc.; or 2) when the Path Vectors are generated from end-to-end measurement data.

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 representation for a component in a network that handles data packets and whose properties can potentially have an impact on the end-to-end performance of traffic. An ANE can be a physical device such as a router, a link or an interface, or an aggregation of devices such as a subnetwork or a data center.

The definition of Abstract Network Element is similar to Network Element defined in [RFC2216] in the sense that they both provide an abstract representation of specific components of a network. However, they have different criteria on how these particular components are selected. Specifically, a Network Element requires the components to be capable of exercising QoS control, while Abstract Network Element only requires the components to have an impact on the end-to-end performance.

ANE Name: A string that uniquely identifies an ANE in a specific

scope. An ANE can be constructed either statically in advance or on demand based on the requested information. Thus, different ANEs may only be valid within a particular scope, either ephemeral or persistent. Within each scope, an ANE is uniquely identified by an ANE Name, as defined in Section 6.1. Note that an ALTO client must not assume ANEs in different scopes but with the same ANE Name refer to the same component(s) of the network.

Path Vector: Path Vector, or ANE Path Vector, refers to a JSON array of ANE Names. It is a generalization of BGP path vector. While standard BGP path vector (Section 5.1.2 of [RFC4271]) specifies a sequence of autonomous systems for a destination IP prefix, the Path Vector defined in this extension specifies a sequence of ANEs either for a source Provider-Defined Identifier (PID) and a destination PID as in the CostMapData (11.2.3.6 in [RFC7285]), or for a source endpoint and a destination endpoint as in the EndpointCostMapData object (Section 11.5.1.6 of [RFC7285]).

Path Vector resource: An ALTO information resource (Section 8.1 of [RFC7285]) which supports the extension defined in this document.

Path Vector cost type: 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 Filtered Cost Map request or an Endpoint Cost Service request, it indicates each cost value must be interpreted as a Path Vector.

Path Vector request: 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.

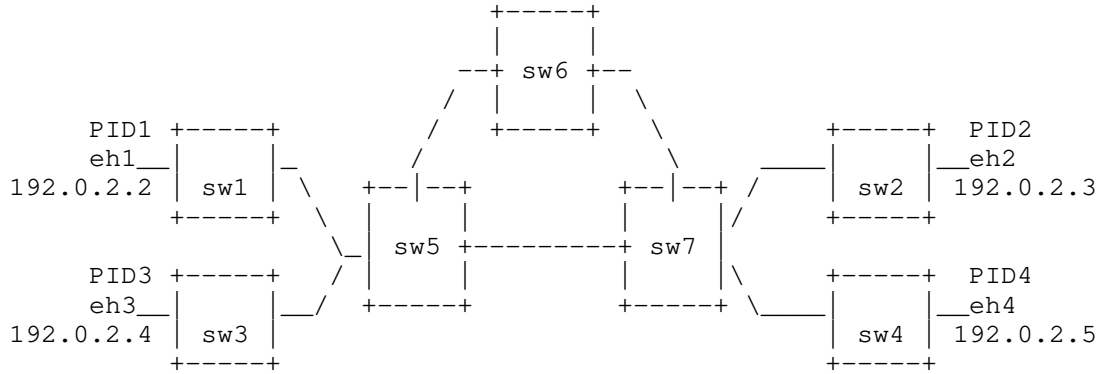
4. Requirements and Use Cases

4.1. Design Requirements

This section gives an illustrative example of how an overlay application can benefit from the extension defined in this document.

Assume that an application has control over a set of flows, which may go through shared links/nodes and share bottlenecks. The application seeks to schedule the traffic among multiple flows to get better performance. The constraints of feasible rate allocations of those flows will benefit the scheduling. However, Cost Maps as defined in [RFC7285] 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", "sw2", "sw3" and "sw4" are access switches, and sw5-sw7 form the backbone. End hosts eh1 to eh4 are connected to access switches sw1 to sw4 respectively. Assume that the bandwidth of link eh1 -> sw1 and link sw1 -> sw5 is 150 Mbps, and the bandwidth of the other links is 100 Mbps.



$\text{bw}(\text{eh1} \rightarrow \text{sw1}) = \text{bw}(\text{sw1} \rightarrow \text{sw5}) = 150 \text{ Mbps}$
 $\text{bw}(\text{eh2} \rightarrow \text{sw2}) = \text{bw}(\text{eh3} \rightarrow \text{sw3}) = \text{bw}(\text{eh4} \rightarrow \text{sw4}) = 100 \text{ Mbps}$
 $\text{bw}(\text{sw1} \rightarrow \text{sw5}) = \text{bw}(\text{sw3} \rightarrow \text{sw5}) = \text{bw}(\text{sw2} \rightarrow \text{sw7}) = \text{bw}(\text{sw4} \rightarrow \text{sw7}) = 100 \text{ Mbps}$
 $\text{bw}(\text{sw5} \rightarrow \text{sw6}) = \text{bw}(\text{sw5} \rightarrow \text{sw7}) = \text{bw}(\text{sw6} \rightarrow \text{sw7}) = 100 \text{ Mbps}$

Figure 1: Raw Network Topology

The base ALTO topology abstraction of the network is shown in Figure 2. Assume the cost map returns an hypothetical cost type representing the available bandwidth between a source and a destination.



Figure 2: Base Topology Abstraction

Now assume the application wants to maximize the total rate of the traffic among a set of <source, destination> pairs, say "eh1 -> eh2" and "eh1 -> eh4". Let "x" denote the transmission rate of "eh1 -> eh2" and "y" denote the rate of "eh1 -> eh4". The objective function is

$$\max(x + y).$$

With the ALTO Cost Map, the cost between PID1 and PID2 and between PID1 and PID4 will both be 100 Mbps. The client can get a capacity region of

$$\begin{aligned} x &\leq 100 \text{ Mbps,} \\ y &\leq 100 \text{ Mbps.} \end{aligned}$$

With this information, the client may mistakenly think it can achieve a maximum total rate of 200 Mbps. However, this rate is infeasible, as there are only two potential cases:

- * Case 1: "eh1 -> eh2" and "eh1 -> eh4" take different path segments from "sw5" to "sw7". For example, if "eh1 -> eh2" uses path "eh1 -> sw1 -> sw5 -> sw6 -> sw7 -> sw2 -> eh2" and "eh1 -> eh4" uses path "eh1 -> sw1 -> sw5 -> sw7 -> sw4 -> eh4", then the shared bottleneck links are "eh1 -> sw1" and "sw1 -> sw5". In this case, the capacity region is:

$$\begin{aligned} x &\leq 100 \text{ Mbps} \\ y &\leq 100 \text{ Mbps} \\ x + y &\leq 150 \text{ Mbps} \end{aligned}$$

and the real optimal total rate is 150 Mbps.

- * Case 2: "eh1 -> eh2" and "eh1 -> eh4" take the same path segment from "sw5" to "sw7". For example, if "eh1 -> eh2" uses path "eh1 -> sw1 -> sw5 -> sw7 -> sw2 -> eh2" and "eh1 -> eh4" also uses path "eh1 -> sw1 -> sw5 -> sw7 -> sw4 -> eh4", then the shared bottleneck link is "sw5 -> sw7". In this case, the capacity region is:

$$\begin{aligned} x &\leq 100 \text{ Mbps} \\ y &\leq 100 \text{ Mbps} \\ x + y &\leq 100 \text{ Mbps} \end{aligned}$$

and the real optimal total rate is 100 Mbps.

Clearly, with more accurate and fine-grained information, the application can gain a better prediction of its traffic and may orchestrate its resources accordingly. However, to provide such information, the network needs to expose abstract information beyond the simple cost map abstraction. In particular:

- * The ALTO server must expose abstract information about the network paths that are traversed by the traffic between a source and a destination beyond a simple numerical value, which allows the overlay application to distinguish between Cases 1 and 2 and to compute the optimal total rate accordingly.
- * The ALTO server must allow the client to distinguish the common ANE shared by "eh1 -> eh2" and "eh1 -> eh4", e.g., "eh1 - sw1" and "sw1 - sw5" in Case 1.
- * The ALTO server must expose abstract information on the properties of the ANEs used by "eh1 -> eh2" and "eh1 -> eh4". For example, an ALTO server can either expose 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, or expose 3 abstract elements "A", "B" and "C", which represent the linear constraints that define the same capacity region 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 the ANEs that are important to assess the QoE of the overlay application on the path of a <source, destination> pair.

AR2: An ALTO server must provide information to identify how ANEs are shared on the paths of different <source, destination> pairs.

AR3: An ALTO server must provide information on the properties that are important to assess the QoE of the application for ANEs.

The extension defined in this document specifies a solution to expose such abstract information.

4.2. Sample Use Cases

While the multiple flow scheduling problem is used to help identify the additional requirements, the extension defined in this document can be applied to a wide range of applications. This section highlights some use cases that are reported.

4.2.1. Exposing Network Bottlenecks

An important use case of the Path Vector extension is to expose network bottlenecks. Applications which need to perform large scale data transfers can benefit from being aware of the resource constraints exposed by this extension even if they have different objectives. One such example is the Worldwide LHC Computing Grid (WLCG), the largest example of a distributed computation collaboration in the research and education world.

Figure 3 illustrates an example of using ALTO Path Vector as an interface between the job optimizer for a data analytics system and the network manager. In particular, we assume the objective of the job optimizer is to minimize the job completion time.

In such a setting, the network-aware job optimizer (e.g., [CLARINET]) takes a query and generates multiple query execution plans (QEP). It can encode the QEPs as Path Vector requests that are sent to an ALTO server. The ALTO server obtains the routing information for the flows in a QEP and finds links, routers, or middleboxes (e.g., a stateful firewall) that can potentially become bottlenecks of the QEP (e.g., see [NOVA] and [G2] for mechanisms to identify bottleneck links under different settings). The resource constraint information is encoded in a Path Vector response and returned to the ALTO client.

With the network resource constraints, the job optimizer may choose the QEP with the optimal job completion time to be executed. It must be noted that the ALTO framework itself does not offer the capability to control the traffic. However, certain network managers may offer ways to enforce resource guarantees, such as on-demand tunnels (e.g., [SWAN]), demand vector (e.g., [HUG], [UNICORN]), etc. The traffic control interfaces and mechanisms are out of the scope of this document.

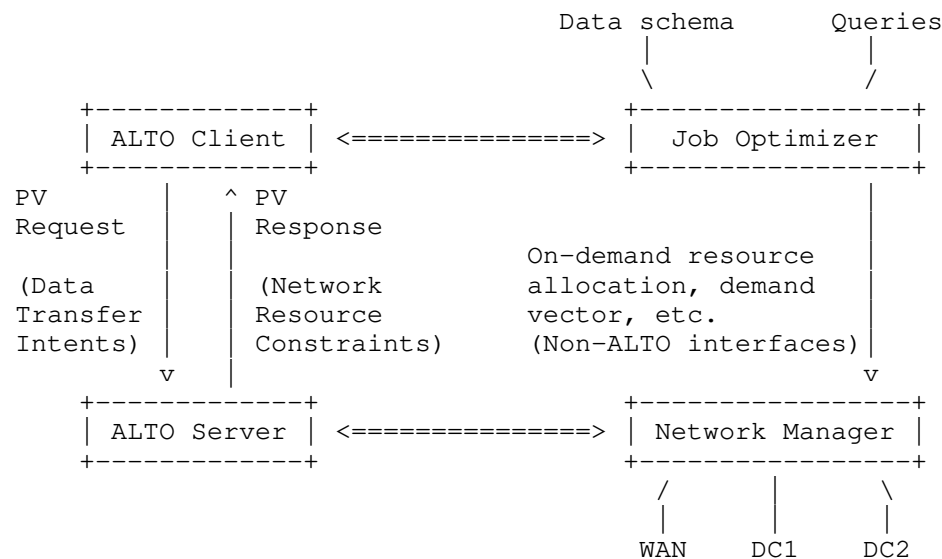


Figure 3: Example Use Case for Data Analytics

Another example is as illustrated in Figure 4. Consider a network consisting of multiple sites and a non-blocking core network, i.e., the links in the core network have sufficient bandwidth that they will not become the bottleneck of the data transfers.

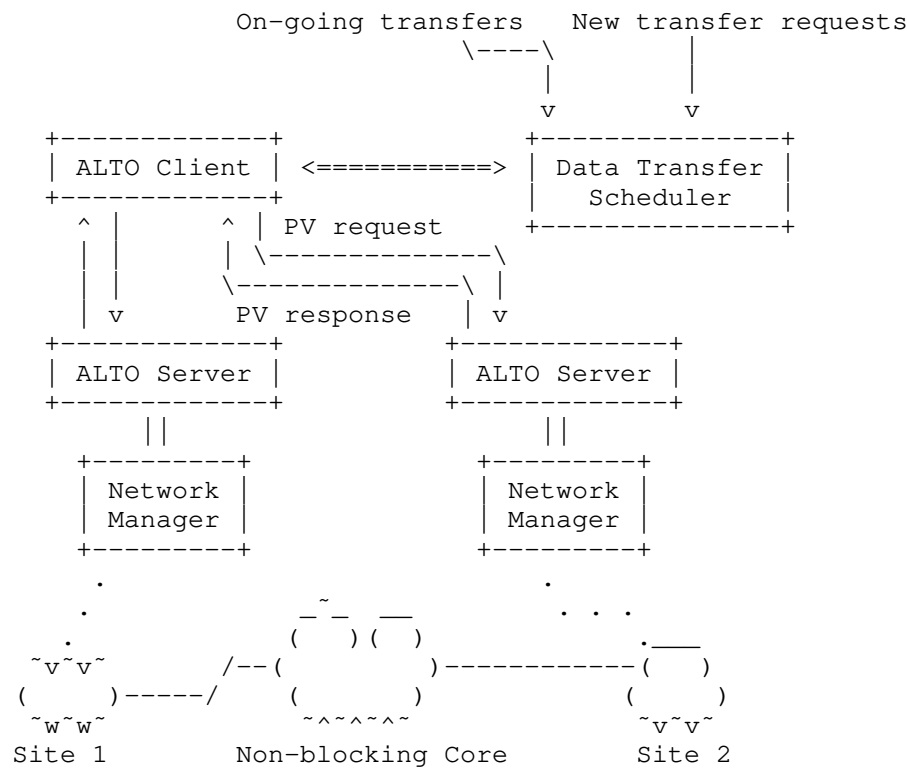
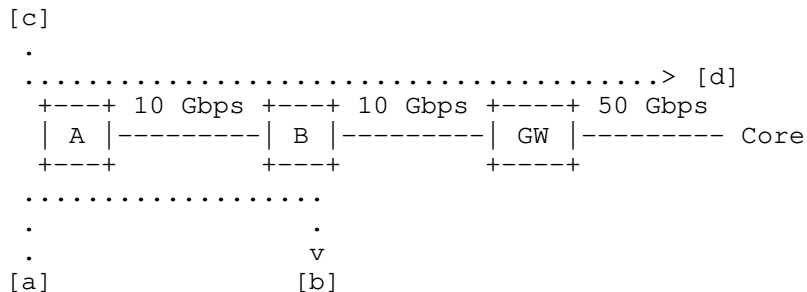


Figure 4: Example Use Case for Cross-site Bottleneck Discovery

Site 1:



Site 2:

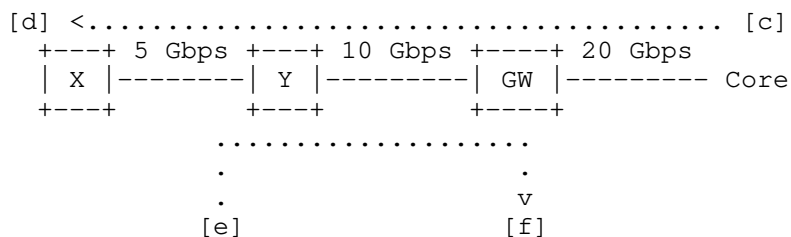


Figure 5: Example: Three Flows in Two Sites

With the Path Vector extension, a site can reveal the bottlenecks inside its own network with necessary information (such as link capacities) to the ALTO client, instead of providing the full topology and routing information, or no bottleneck information at all. The bottleneck information can be used to analyze the impact of adding/removing data transfer flows, e.g., using the [G2] framework. For example, assume hosts "a", "b", "c" are in site 1 and hosts "d", "e", "f" are in site 2, and there are 3 flows in two sites: "a -> b", "c -> d", "e -> f". For these flows, site 1 returns:

```

a: { b: [ane1] },
c: { d: [ane1, ane2, ane3] }

ane1: bw = 10 Gbps (link: A->B)
ane2: bw = 10 Gbps (link: B->GW)
ane3: bw = 50 Gbps (link: GW->Core)

```

and site 2 returns:

```
c: { d: [anei, aneii, aneiii] }  
e: { f: [aneiv] }
```

```
anei: bw = 5 Gbps (link Y->X)  
aneii: bw = 10 Gbps (link GW->Y)  
aneiii: bw = 20 Gbps (link Core->GW)  
aneiv: bw = 10 Gbps (link Y->GW)
```

With the information, the data transfer scheduler can use algorithms such as the theory on bottleneck structure [G2] to predict the potential throughput of the flows.

4.2.2. Resource Exposure for CDN and Service Edge

A growing trend in today's applications (2021) is to bring storage and computation closer to the end users for better QoE, such as Content Delivery Network (CDN), AR/VR, and cloud gaming, as reported in various documents (e.g., [SEREDGE] and [MOWIE]). Internet Service Providers may deploy multiple layers of CDN caches, or more generally service edges, with different latency and available resources including number of CPU cores, memory, and storage.

For example, Figure 6 illustrates a typical edge-cloud scenario where memory is measured in Gigabytes (G) and storage is measured in Terabytes (T). The "on-premise" edge nodes are closest to the end hosts and have the smallest latency, and the site-radio edge node and access central office (CO) have larger latency but more available resources.

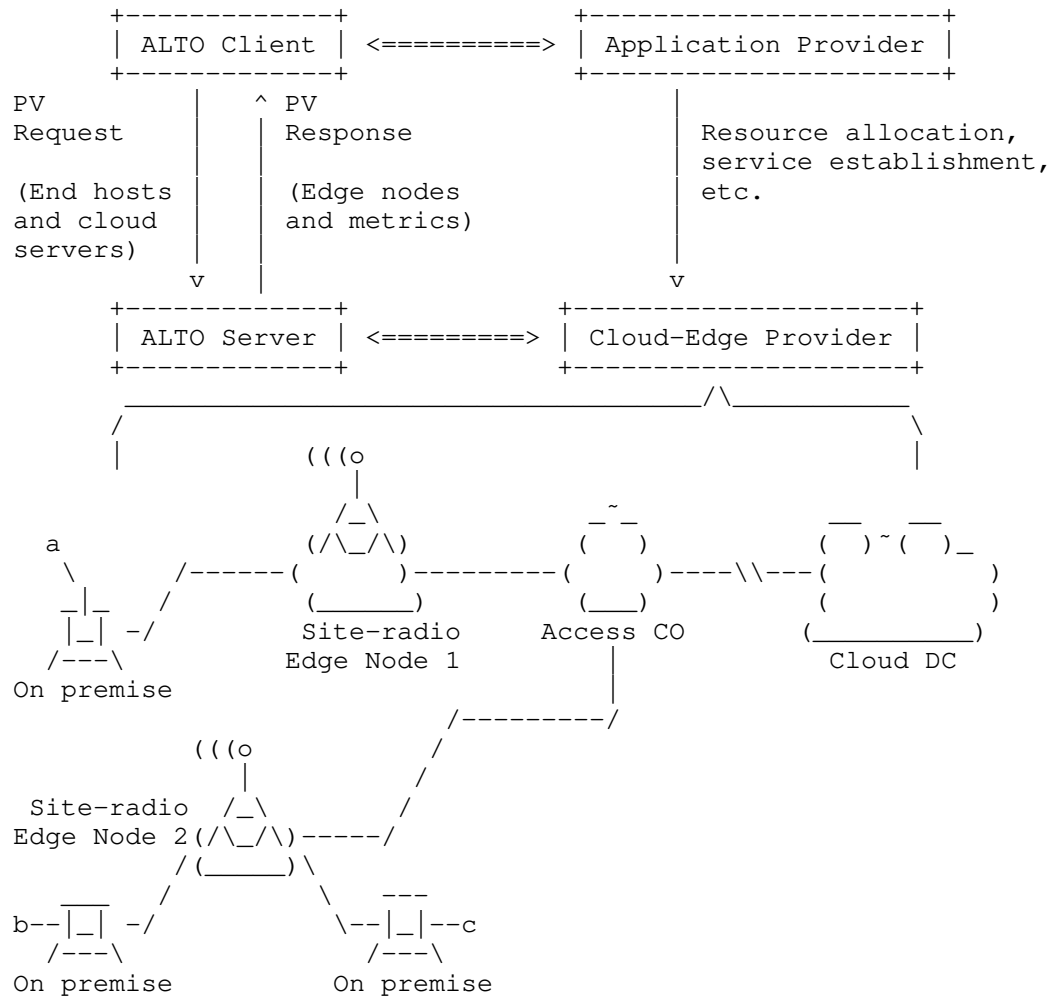


Figure 6: Example Use Case for Service Edge Exposure

```
a: { b: [ane1, ane2, ane3, ane4, ane5],  
      c: [ane1, ane2, ane3, ane4, ane6],  
      DC: [ane1, ane2, ane3] }  
b: { c: [ane5, ane4, ane6], DC: [ane5, ane4, ane3] }  
  
ane1: latency=5ms cpu=2 memory=8G storage=10T  
(on premise, a)  
  
ane2: latency=20ms cpu=4 memory=8G storage=10T  
(Site-radio Edge Node 1)  
  
ane3: latency=100ms cpu=8 memory=128G storage=100T  
(Access CO)  
  
ane4: latency=20ms cpu=4 memory=8G storage=10T  
(Site-radio Edge Node 2)  
  
ane5: latency=5ms cpu=2 memory=8G storage=10T  
(on premise, b)  
  
ane6: latency=5ms cpu=2 memory=8G storage=10T  
(on premise, c)
```

Figure 7: Example Service Edge Query Results

With the extension defined in this document, an ALTO server can selectively reveal the CDNs and service edges that reside along the paths between different end hosts and/or the cloud servers, together with their properties such as capabilities (e.g., storage, GPU) and available Service Level Agreement (SLA) plans. See Figure 7 for an example where the query is made for sources [a, b] and destinations [b, c, DC]. Here each ANE represents a service edge and the properties include access latency, available resources, etc. Note the properties here are only used for illustration purposes and are not part of this extension.

With the service edge information, an ALTO client may better conduct CDN request routing or offload functionalities from the user equipment to the service edge, with considerations on customized quality of experience.

5. Path Vector Extension: Overview

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

To satisfy the additional requirements listed in Section 4.1, this extension:

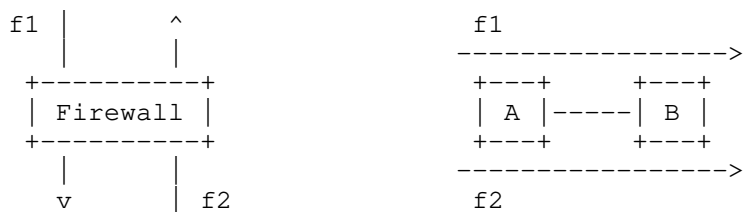
1. introduces the concept of Abstract Network Element (ANE) as the abstraction of components in a network whose properties may have an impact on the end-to-end performance of the traffic handled by those components,
2. extends the Cost Map and Endpoint Cost Service to convey the ANEs traversed by the path of a <source, destination> pair as Path Vectors, and
3. uses the Unified Property Map to convey the association between the ANEs and their properties.

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

5.1. Abstract Network Element (ANE)

This extension introduces ANE as an indirect and network-agnostic way to specify a component or an aggregation of components of a network whose properties have an impact on the end-to-end performance for application traffic between endpoints.

ANEs allow ALTO servers to focus on common properties of different types of network components. For example, the throughput of a flow can be constrained by different components in a network: the capacity of a physical link, the maximum throughput of a firewall, the reserved bandwidth of an MPLS tunnel, etc. See the example below, assume the throughput of the firewall is 100 Mbps and the capacity for link (A, B) is also 100 Mbps, they result in the same constraint on the total throughput of f1 and f2. Thus, they are identical when treated as an ANE.



When an ANE is defined by an ALTO server, it is assigned an identifier by the ALTO server, i.e., a string of type ANEName as specified in Section 6.1, and a set of associated properties.

5.1.1. ANE Entity Domain

In this extension, the associations between ANE and the properties are conveyed in a Unified Property Map. Thus, ANEs must constitute an entity domain (Section 5.1 of [I-D.ietf-alto-unified-props-new]), and each ANE property must be an entity property (Section 5.2 of [I-D.ietf-alto-unified-props-new]).

Specifically, this document defines a new entity domain called "ane" as specified in Section 6.2 and defines two initial properties for the ANE entity domain.

5.1.2. Ephemeral and Persistent ANEs

By design, ANEs are ephemeral and not to be used in further requests to other ALTO resources. More precisely, the corresponding ANE names are no longer valid beyond the scope of a Path Vector response or the incremental update stream for a Path Vector request. Compared with globally unique ANE names, ephemeral ANE has several benefits including better privacy of the ISP's internal structure and more flexible ANE computation.

For example, an ALTO server may define an ANE for each aggregated bottleneck link between the sources and destinations specified in the request. For requests with different sources and destinations, the bottlenecks may be different but can safely reuse the same ANE names. The client can still adjust its traffic based on the information but is difficult to infer the underlying topology with multiple queries.

However, sometimes an ISP may intend to selectively reveal some "persistent" network components which, opposite to being ephemeral, have a longer life cycle. For example, an ALTO server may define an ANE for each service edge cluster. Once a client chooses to use a service edge, e.g., by deploying some user-defined functions, it may want to stick to the service edge to avoid the complexity of state transition or synchronization, and continuously query the properties of the edge cluster.

This document provides a mechanism to expose such network components as persistent ANEs. A persistent ANE has a persistent ID that is registered in a Property Map, together with their properties. See Section 6.2.4 and Section 6.4.2 for more detailed instructions on how to identify ephemeral ANEs and persistent ANEs.

5.1.3. Property Filtering

Resource-constrained ALTO clients (see Section 4.1.2 of [RFC7285]) 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 properties selected by a client as being of interest are specified in the subsequent Path Vector queries using the filter called 'ane-property-names'. The response includes and only includes the selected properties for the ANEs in the response.

The "ane-property-names" capability for Cost Map and for Endpoint Cost Service is specified in Section 7.2.4 and Section 7.3.4 respectively. The "ane-property-names" filter for Cost Map and Endpoint Cost Service is specified in Section 7.2.3 and Section 7.3.3 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.

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 C++ where there existed a JSON array type named JSONArray, 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", it means that 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 8.

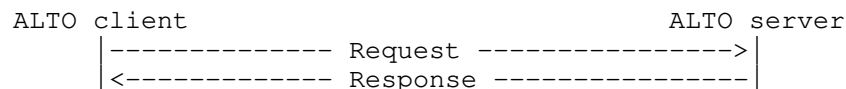


Figure 8: A Typical ALTO Request and Response

The extension defined in this document, on the other hand, involves two types of information resources: Path Vectors conveyed in an `InfoResourceCostMap` (defined in Section 11.2.3.6 of [RFC7285]) or an `InfoResourceEndpointCostMap` (defined in Section 11.5.1.6 of [RFC7285]), and ANE properties conveyed in an `InfoResourceProperties` (defined in Section 7.6 of [I-D.ietf-alto-unified-props-new]).

Instead of two consecutive message exchanges, the extension defined in this document enforces one round of communication. Specifically, the ALTO client must include the source and destination pairs and the requested ANE properties in a single request, and the ALTO server must return a single response containing both the Path Vectors and properties associated with the ANEs in the Path Vectors, as shown in Figure 9. Since the two parts are bundled together in one response message, their orders are interchangeable. See Section 7.2.6 and Section 7.3.6 for details.

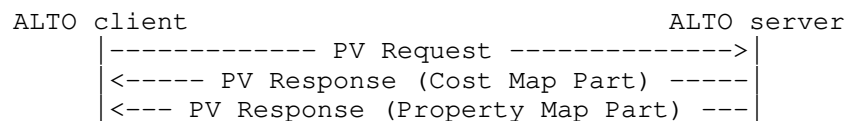


Figure 9: The Path Vector Extension Request and Response

This design is based on the following considerations:

1. 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 in an InfoResourceCostMap or an InfoResourceEndpointCostMap, and the Property Map is encoded in an InfoResourceProperties. 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 Service). 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" field in the IRD entry is "multipart/related" with the parameter "type=application/alto-costmap+json"; for an Endpoint Cost Service, the parameter is "type=application/alto-endpointcost+json".

5.3.2. References to Part Messages

As the response of a Path Vector resource is a multipart message with two different parts, it is important that each part can be uniquely identified. Following the designs of [RFC8895], this extension requires that an ALTO server assigns a unique identifier to each part of the multipart response message. This identifier, referred to as a Part Resource ID (See Section 6.6 for details), is present in the part message's "Content-ID" header. By concatenating the Part Resource ID to the identifier of the Path Vector request, an ALTO server/client can uniquely identify the Path Vector Part or the

Property Map part.

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 Entity Domain

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

It must be noted that the term "domain" here does not refer to a network domain. Rather, it is inherited from the "entity domain" defined in Sec 3.2 in [I-D.ietf-alto-unified-props-new] that represents the set of valid entities defined by an ALTO information resource (called the defining information resource).

6.2.1. Entity Domain Type

The Entity Domain Type is "ane".

6.2.2. Domain-Specific Entity Identifier

The entity identifiers are the ANE Names in the associated Property Map.

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

The defining resource for entity domain type "ane" MUST be a Property Map, i.e., the media type of defining resources is:

application/alto-propmap+json

Specifically, for ephemeral ANEs that appear in a Path Vector response, their entity domain names MUST be exactly ".ane" and the defining resource of these ANEs is the Property Map part of the

multipart response. Meanwhile, for any persistent ANE whose defining resource is a Property Map resource, its entity domain name MUST have the format of "PROPMAP.ane" where PROPMAP is the resource ID of the defining resource. Persistent entities are "persistent" because standalone queries can be made by an ALTO client to their defining resource(s) when the connection to the Path Vector service is closed.

For example, the defining resource of an ephemeral ANE whose entity identifier is ".ane:NET1" is the Property Map part that contains this identifier. The defining resource of a persistent ANE whose entity identifier is "dc-props.ane:DC1" is the Property Map with the resource ID "dc-props".

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

Two initial ANE property types are specified, "max-reservable-bandwidth" and "persistent-entity-id".

Note that these property types do not depend on any information resource. As such, the EntityPropertyName MUST only have the EntityPropertyType part.

6.4.1. Maximum Reservable Bandwidth

The maximum reservable bandwidth property ("max-reservable-bandwidth") 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 (bps). If this property is requested by the ALTO client but not present for an ANE in the server response, it MUST be interpreted as that the property is not defined for the ANE.

This property can be offered in a setting where the ALTO server is part of a network system that provides on-demand resource allocation and the ALTO client is part of a user application. One existing example is [NOVA]: the ALTO server is part of an SDN controller and exposes a list of traversed network elements and associated link bandwidth to the client. The encoding in [NOVA] differs from the Path Vector response defined in this document that the Path Vector part and Property Map part are put in the same JSON object.

In such a framework, the ALTO server exposes resource (e.g., reservable bandwidth) availability information to the ALTO client. How the client makes resource requests based on the information and how the resource allocation is achieved respectively depend on interfaces between the management system and the users or a higher-layer protocol (e.g., SDN network intents or MPLS tunnels), which are out of the scope of this document.

6.4.2. Persistent Entity ID

The persistent entity ID property is the entity identifier of the persistent ANE which an ephemeral ANE presents (See Section 5.1.2 for details). The value of this property is encoded with the format EntityID defined in Section 5.1.3 of [I-D.ietf-alto-unified-props-new].

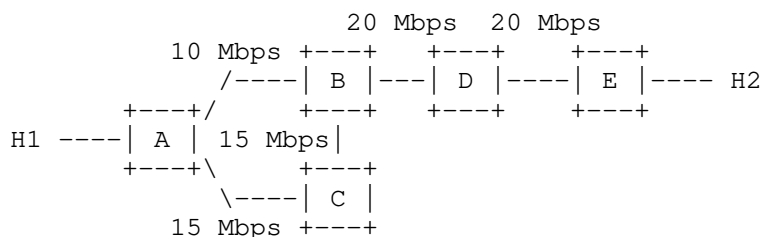
In this format, the entity ID combines:

- * a defining information resource for the ANE on which a "persistent-entity-id" is queried, which is the Property Map resource defining the ANE as a persistent entity, together with the properties;
- * the persistent name of the ANE in that Property Map.

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

6.4.3. Examples

To illustrate the use of "max-reservable-bandwidth", consider the following network with 5 nodes. Assume the client wants to query the maximum reservable bandwidth from H1 to H2. An ALTO server may split the network into two ANEs: "ane1" that represents the subnetwork with routers A, B, and C, and "ane2" that represents the subnetwork with routers B, D and E. The maximum reservable bandwidth for "ane1" is 15 Mbps (using path A->C->B) and the maximum reservable bandwidth for "ane2" is 20 Mbps (using path B->D->E).



To illustrate the use of "persistent-entity-id", consider the scenario in Figure 6. As the life cycle of service edges are typically long, they may contain information that is not specific to the query. Such information can be stored in an individual unified property map and later be accessed by an ALTO client.

For example, "ane1" in Figure 7 represents the on-premise service edge closest to host a. Assume the properties of the service edges are provided in a unified property map called "se-props" and the ID of the on-premise service edge is "9a0b55f7-7442-4d56-8a2c-b4cc6a8e3aa1", the "persistent-entity-id" of "ane1" will be "se-props.ane:9a0b55f7-7442-4d56-8a2c-b4cc6a8e3aa1". With this persistent entity ID, an ALTO client may send queries to the "se-props" resource with the entity ID ".ane:9a0b55f7-7442-4d56-8a2c-b4cc6a8e3aa1".

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 extension defined in this document.

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.

An ALTO client MUST interpret the Path Vector as if the traffic between a source and a destination logically traverses the ANEs in the same order as they appear in the Path Vector.

When the Path Vector procedures defined in this document are in use, an ALTO server using the "ane-path" cost metric and the "array" cost mode (see Section 6.5.2) MUST return as the cost value a JSON array of ANEName and the client MUST also check that each element contained in the array is an ANEName (Section 6.1). Otherwise, the client MUST discard the response and SHOULD follow the instructions in Section 8.3.4.3 of [RFC7285] to handle the error.

6.5.2. Cost Mode: array

The cost mode "array" indicates that every cost value in the response body of a (Filtered) Cost Map or an Endpoint Cost Service MUST be interpreted as a JSON array object. While this cost mode can be applied to all cost metrics, additional specifications will be needed to clarify the semantics of the array cost mode when combined with cost metrics other than 'ane-path'.

6.6. Part Resource ID and Part Content 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]).

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.

A Part Content ID conforms to the format of msg-id as specified in [RFC2387] and [RFC5322]. Specifically, it has the following format:

"<" PART-RESOURCE-ID "@" DOMAIN-NAME ">"

PART-RESOURCE-ID: PART-RESOURCE-ID has the same format as the Part Resource ID. It is used to identify whether a part message is a Path Vector or a Property Map.

DOMAIN-NAME: DOMAIN-NAME has the same format as dot-atom-text specified in Section 3.2.3 of [RFC5322]. It must be the domain name of the ALTO server.

7. Specification: Service Extensions

7.1. Notations

This document uses the same syntax and notations as introduced in Section 8.2 of RFC 7285 [RFC7285] to specify the extensions to existing ALTO resources and services.

7.2. 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 with the Cost Map resource in the same response.

7.2.1. Media Type

The media type of the multipart Filtered Cost Map resource is "multipart/related" and the required "type" parameter MUST have a value of "application/alto-costmap+json".

7.2.2. HTTP Method

The multipart Filtered Cost Map is requested using the HTTP POST method.

7.2.3. 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, which is defined as a JSON object of type ReqFilteredCostMap in Section 4.1.2 of RFC 8189 [RFC8189], with a data format indicated by the media type "application/alto-costmapfilter+json", which is a JSON object of type PVReqFilteredCostMap:

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

with fields:

ane-property-names: A list of selected ANE properties to be included in the response. Each property in this list MUST match one of the supported ANE properties indicated in the resource's "ane-property-names" capability (Section 7.2.4). If the field is not present, it MUST be interpreted as an empty list.

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: 201
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" ]
}
```

7.2.4. Capabilities

The multipart Filtered Cost Map resource extends the capabilities defined in Section 4.1.1 of [RFC8189]. The capabilities are defined by a JSON object of type PVFilteredCostMapCapabilities:

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

with fields:

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.

This extension also introduces additional restrictions for the following fields:

cost-type-names: The "cost-type-names" field MUST 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 (Section 4.1.1 of [RFC8189]): If the "cost-type-names" field includes the Path Vector cost type and the "testable-cost-type-names" field is present, the Path Vector cost type MUST NOT be included in the "testable-cost-type-names" field unless specifically instructed by a future document.

7.2.5. 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.

7.2.6. 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 [RFC2387] with the following parameters:

type: The type parameter is mandatory and 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] and is optional. If present, it MUST have the same value as the "Content-ID" header of the Path Vector part.

boundary: The boundary parameter is as defined in Section 5.1.1 of [RFC2046] and is mandatory.

The body of the response MUST consist of two parts:

- * The Path Vector part MUST include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-costmap+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

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] when the "cost-type" field is present in the input parameters and MUST be a JSON object with the same format as defined in Section 4.1.3 of [RFC8189] if the "multi-cost-types" field is present. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned CostMapData. The resource ID of the version tag MUST follow the format of

resource-id '.' part-resource-id

where "resource-id" is the resource Id of the Path Vector resource, and "part-resource-id" has the same value as the PART-RESOURCE-ID in the "Content-ID" of the Path Vector part. 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 "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-propmap+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

The body of the Unified Property Map part is a JSON object with the same format as defined in Section 7.6 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 for each ANE has one member for each property that is both 1) associated with the ANE, and 2) specified in the "ane-property-names" in the request. If the Path Vector cost type is not included in the "cost-type" field or the "multi-cost-type" field, the "property-map" field MUST be present and the value MUST be an empty object ({}).

A complete and valid response MUST include both the Path Vector part and the Property Map part in the multipart message. If any part is NOT present, the client MUST discard the received information and send another request if necessary.

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. When doing so, an ALTO server MAY choose not to set the "start" parameter, which implies the first part is the root object.

Example: Consider the network in Figure 1. The response of the example request in Section 7.2.3 is as follows, where "ANE1" represents the aggregation of all the switches in the network.

```
HTTP/1.1 200 OK
Content-Length: 859
Content-Type: multipart/related; boundary=example-1;
              type=application/alto-costmap+json

--example-1
Content-ID: <costmap@alto.example.com>
Content-Type: application/alto-costmap+json

{
  "meta": {
    "vtag": {
      "resource-id": "filtered-cost-map-pv.costmap",
      "tag": "fb20b76204814e9db37a51151faaaef2"
    },
    "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
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "filtered-cost-map-pv.costmap",
        "tag": "fb20b76204814e9db37a51151faaaef2"
      }
    ]
  },
  "property-map": {
    ".ane:ANE1": { "max-reservable-bandwidth": 100000000 }
  }
}
```

7.3. Multipart Endpoint Cost Service for Path Vector

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

7.3.1. Media Type

The media type of the multipart Endpoint Cost Service resource is "multipart/related" and the required "type" parameter MUST have a value of "application/alto-endpointcost+json".

7.3.2. HTTP Method

The multipart Endpoint Cost Service resource is requested using the HTTP POST method.

7.3.3. Accept Input Parameters

The input parameters of the multipart Endpoint Cost Service resource are supplied in the body of an HTTP POST request. This document extends the input parameters to an Endpoint Cost Service, which is defined as a JSON object of type ReqEndpointCost in Section 4.2.2 of [RFC8189], with a data format indicated by the media type "application/alto-endpointcostparams+json", which is a JSON object of type PVReqEndpointCost:

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

with fields:

ane-property-names: This document defines the "ane-property-names" in PVReqEndpointcost as the same as in PVReqFilteredCostMap. See Section 7.2.3.

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: 227
Content-Type: application/alto-endpointcostparams+json
```

```
{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [ "ipv4:192.0.2.18" ]
  },
  "ane-property-names": [ "max-reservable-bandwidth" ]
}
```

7.3.4. Capabilities

The capabilities of the multipart Endpoint Cost Service resource are defined by a JSON object of type `PVEndpointcostCapabilities`, which is defined as the same as `PVFilteredCostMapCapabilities`. See Section 7.2.4.

7.3.5. Uses

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

7.3.6. 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" and is mandatory.

start: The start parameter is as defined in Section 7.2.6.

boundary: The boundary parameter is as defined in Section 5.1.1 of [RFC2046] and is mandatory.

The body MUST consist of two parts:

- * The Path Vector part MUST include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-endpointcost+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

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] when the "cost-type" field is present in the input parameters and MUST be a JSON object with the same format as defined in Section 4.2.3 of [RFC8189] if the "multi-cost-types" field is present. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned EndpointCostMapData. The resource ID of the version tag MUST follow the format of

resource-id '.' part-resource-id

where "resource-id" is the resource Id of the Path Vector resource, and "part-resource-id" has the same value as the PART-RESOURCE-ID in the "Content-ID" of the Path Vector part.

- * The Unified Property Map part MUST also include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-propmap+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

The body of the Unified Property Map part MUST be a JSON object with the same format as defined in Section 7.6 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 for each ANE has one member for each property that is both 1) associated with the ANE, and 2) specified in the "ane-property-names" in the request. If the Path Vector cost type is not included in the "cost-type" field or the "multi-cost-type" field, the "property-map" field MUST be present and the value MUST be an empty object ({}).

A complete and valid response MUST include both the Path Vector part and the Property Map part in the multipart message. If any part is NOT present, the client MUST discard the received information and send another request if necessary.

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. When doing so, an ALTO server MAY choose not to set the "start" parameter, which implies the first part is the root object.

Example: Consider the network in Figure 1. The response of the example request in Section 7.3.3 is as follows.

```
HTTP/1.1 200 OK
Content-Length: 845
Content-Type: multipart/related; boundary=example-1;
             type=application/alto-endpointcost+json
```

```
--example-1
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "vtag": {
      "resource-id": "ecs-pv.ecs",
      "tag": "ec137bb78118468c853d5b622ac003f1"
    },
    "dependent-vtags": [
      {
        "resource-id": "my-default-networkmap",
        "tag": "677fe5f4066848d282ece213a84f9429"
      }
    ],
    "cost-type": { "cost-mode": "array", "cost-metric": "ane-path" },
    "cost-map": {
      "ipv4:192.0.2.2": { "ipv4:192.0.2.18": ["ANE1"] }
    }
  }
}
```

```
--example-1
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "ecs-pv.ecs",
        "tag": "ec137bb78118468c853d5b622ac003f1"
      }
    ]
  },
  "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. Sample Setup

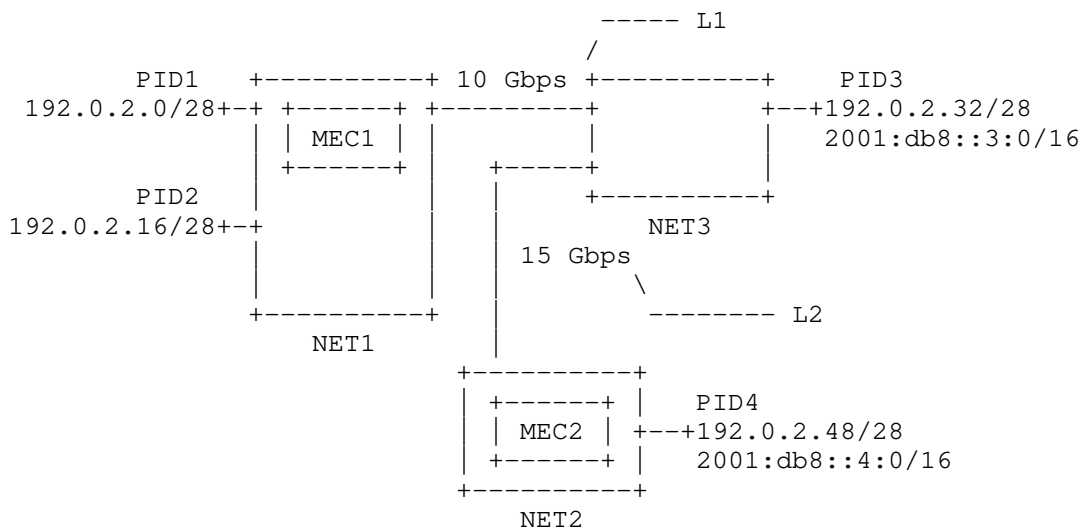


Figure 10: Examples of ANE Properties

In this document, Figure 10 is used to illustrate the message contents. 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.

8.2. Information Resource Directory

To give a comprehensive example of the extension defined in this document, we consider the network in Figure 10. Assume that the 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.
- * "multicost-pv": A Multipart Endpoint Cost Service with both Multi-Cost and Path Vector.

Below is the Information Resource Directory of the example ALTO server. To enable the extension defined in this document, 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 "filtered-cost-map-pv" and "endpoint-cost-pv".

```
{
  "meta": {
    "cost-types": {
      "path-vector": {
        "cost-mode": "array",
        "cost-metric": "ane-path"
      },
      "num-rc": {
        "cost-mode": "numerical",
        "cost-metric": "routingcost"
      }
    }
  },
  "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
  }
},
"multicost-pv": {
  "uri": "https://alto.exmaple.com/endpointcost/mcpv",
  "media-type": "multipart/related;
    type=application/alto-endpointcost+json",
  "accepts": "application/alto-endpointcostparams+json",
  "capabilities": {
    "cost-type-names": [ "path-vector", "num-rc" ],
    "max-cost-types": 2,
    "testable-cost-type-names": [ "num-rc" ],
    "ane-property-names": [
      "max-reservable-bandwidth", "persistent-entity-id"
    ]
  },
  "uses": [ "ane-props" ]
}
}
```

```
}
```

8.3. 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: 153
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: 855
Content-Type: multipart/related; boundary=example-1;
             type=application/alto-costmap+json
```

```
--example-1
Content-ID: <costmap@alto.example.com>
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": "c04bc5da49534274a6daeee8ea1dec62"
      }
    ],
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"
    }
  },
  "cost-map": {
    "PID1": {
      "PID3": [ "L1" ],
      "PID4": [ "L1", "L2" ]
    }
  }
}
--example-1
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

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

```

8.4. Multipart Endpoint Cost Service 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 for two IPv4 source and destination pairs (192.0.2.34 -> 192.0.2.2 and 192.0.2.34 -> 192.0.2.50) and one IPv6 source and destination pair (2001:db8::3:1 -> 2001:db8::4:1).

The response consists of two parts. The first part returns the array of ANENAME for each valid source and destination pair. As one can see in Figure 10, flow 192.0.2.34 -> 192.0.2.2 traverses NET2, L1 and NET1, and flows 192.0.2.34 -> 192.0.2.50 and 2001:db8::3:1 -> 2001:db8::4:1 traverse NET2, L2 and NET3.

The second part returns the requested properties of ANEs. Assume NET1, NET2 and NET3 has sufficient bandwidth and their "max-reservable-bandwidth" values are set to a sufficiently large number (50 Gbps in this case). On the other hand, assume there are no prior reservation on L1 and L2, and their "max-reservable-bandwidth" values are the corresponding link capacity (10 Gbps for L1 and 15 Gbps for L2).

Both NET1 and NET2 have a mobile edge deployed, i.e., MEC1 in NET1 and MEC2 in NET2. Assume the ANENAME for MEC1 and MEC2 are "MEC1" and "MEC2" and their properties can be retrieved from the Property Map "ane-props". Thus, the "persistent-entity-id" property of NET1 and NET3 are "ane-props.ane:MEC1" and "ane-props.ane:MEC2" respectively.

```
POST /endpointcost/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;
       type=application/alto-endpointcost+json,
       application/alto-error+json
Content-Length: 362
Content-Type: application/alto-endpointcostparams+json
```

```
{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.34",
      "ipv6:2001:db8::3:1"
    ],
    "dsts": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.50",
      "ipv6:2001:db8::4:1"
    ]
  },
  "ane-property-names": [
    "max-reservable-bandwidth",
    "persistent-entity-id"
  ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 1432
Content-Type: multipart/related; boundary=example-2;
             type=application/alto-endpointcost+json
```

```
--example-2
Content-ID: <ecs@alto.example.com>
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": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2": [ "NET3", "L1", "NET1" ],
      "ipv4:192.0.2.50": [ "NET3", "L2", "NET2" ]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [ "NET3", "L2", "NET2" ]
    }
  }
}
--example-2
Content-ID: <propmap@alto.example.com>
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:MEC1"
    },
    ".ane:NET2": {
      "max-reservable-bandwidth": 50000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    },
    ".ane:L1": {
      "max-reservable-bandwidth": 10000000000
    },
    ".ane:L2": {
      "max-reservable-bandwidth": 15000000000
    }
  }
}
```

```
}
```

Under certain scenarios where the traversal order is not crucial, an ALTO server implementation may choose to not follow strictly the physical traversal order and may even obfuscate the order intentionally to preserve its own privacy or conform to its own policies. For example, an ALTO server may choose to aggregate NET1 and L1 as a new ANE with ANE name "AGGR1", and aggregate NET2 and L2 as a new ANE with ANE name "AGGR2". The "max-reservable-bandwidth" of "AGGR1" takes the value of L1, which is smaller than that of NET1, and the "persistent-entity-id" of "AGGR1" takes the value of NET1. The properties of "AGGR2" are computed in a similar way and the obfuscated response is as shown below. Note that the obfuscation of Path Vector responses is implementation-specific and is out of the scope of this document, and developers may refer to Section 11 for further references.

```
HTTP/1.1 200 OK
Content-Length: 1263
Content-Type: multipart/related; boundary=example-2;
              type=application/alto-endpointcost+json
```

```
--example-2
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "vtags": {
      "resource-id": "endpoint-cost-pv.ecs",
      "tag": "bb975862fbe3422abf4dae386b132c1d"
    },
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2": [ "NET3", "AGGR1" ],
      "ipv4:192.0.2.50": [ "NET3", "AGGR2" ]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [ "NET3", "AGGR2" ]
    }
  }
}
```

```
--example-2
```

Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

```
{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "endpoint-cost-pv.ecs",
        "tag": "bb975862fbe3422abf4dae386b132c1d"
      },
      {
        "resource-id": "ane-props",
        "tag": "bf3c8c1819d2421c9a95a9d02af557a3"
      }
    ]
  },
  "property-map": {
    ".ane:AGGR1": {
      "max-reservable-bandwidth": 10000000000,
      "persistent-entity-id": "ane-props.ane:MEC1"
    },
    ".ane:AGGR2": {
      "max-reservable-bandwidth": 15000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    }
  }
}
```

8.5. Incremental Updates

In this example, an ALTO client subscribes to the incremental update for the multipart Endpoint Cost Service 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: 112
```

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

Based on the server-side process defined in [RFC8895], 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=example-3;
      type=application/alto-endpointcost+json,ecspvsub1
data: --example-3
data: Content-ID: <ecsmap@alto.example.com>
data: Content-Type: application/alto-endpointcost+json
data:
data: <endpoint-cost-map-entry>
data: --example-3
data: Content-ID: <propmap@alto.example.com>
data: Content-Type: application/alto-propmap+json
data:
data: <property-map-entry>
data: --example-3--
```

When the contents change, the ALTO server will publish the updates for each node in this tree separately, based on Section 6.7.3 of [RFC8895].

```
event: application/merge-patch+json, ecspvsubl.ecsmap@alto.example.com  
data: <Merge patch for endpoint-cost-map-update>
```

```
event: application/merge-patch+json, ecspvsubl.propmap@alto.example.com  
data: <Merge patch for property-map-update>
```

8.6. Multi-cost

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

The request asks for two cost types: the first is the Path Vector cost type, and the second is a numerical routing cost. It also queries the Maximum Reservable Bandwidth ANE property and the Persistent Entity property for two IPv4 source and destination pairs (192.0.2.34 -> 192.0.2.2 and 192.0.2.34 -> 192.0.2.50) and one IPv6 source and destination pair (2001:db8::3:1 -> 2001:db8::4:1).

The response consists of two parts. The first part returns a JSONArray that contains two JSONValue for each requested source and destination pair: the first JSONValue is a JSONArray of ANENames, which is the value of the Path Vector cost type, and the second JSONValue is a JSONNumber which is the value of the routing cost. The second part contains a Property Map that maps the ANEs to their requested properties.

```
POST /endpointcost/mcpv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;
       type=application/alto-endpointcost+json,
       application/alto-error+json
Content-Length: 433
Content-Type: application/alto-endpointcostparams+json

{
  "multi-cost-types": [
    { "cost-mode": "array", "cost-metric": "ane-path" },
    { "cost-mode": "numerical", "cost-metric": "routingcost" }
  ],
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.34",
      "ipv6:2001:db8::3:1"
    ],
    "dsts": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.50",
      "ipv6:2001:db8::4:1"
    ]
  },
  "ane-property-names": [
    "max-reservable-bandwidth",
    "persistent-entity-id"
  ]
}

HTTP/1.1 200 OK
Content-Length: 1350
Content-Type: multipart/related; boundary=example-4;
             type=application/alto-endpointcost+json

--example-4
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json

{
  "meta": {
    "vtags": {
      "resource-id": "endpoint-cost-pv.ecs",
      "tag": "84a4f9c14f9341f0983e3e5f43a371c8"
    },
    "multi-cost-types": [
      { "cost-mode": "array", "cost-metric": "ane-path" },
      { "cost-mode": "numerical", "cost-metric": "routingcost" }
    ]
  }
}
```



```

    ]
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2": [[ "NET3", "AGGR1" ], 3],
      "ipv4:192.0.2.50": [[ "NET3", "AGGR2" ], 2]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [[ "NET3", "AGGR2" ], 2]
    }
  }
}
--example-4
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "endpoint-cost-pv.ecs",
        "tag": "84a4f9c14f9341f0983e3e5f43a371c8"
      },
      {
        "resource-id": "ane-props",
        "tag": "be157afa031443a187b60bb80a86b233"
      }
    ]
  },
  "property-map": {
    ".ane:AGGR1": {
      "max-reservable-bandwidth": 10000000000,
      "persistent-entity-id": "ane-props.ane:MEC1"
    },
    ".ane:AGGR2": {
      "max-reservable-bandwidth": 15000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    }
  }
}

```

9. Compatibility with Other ALTO Extensions

9.1. Compatibility with Legacy ALTO Clients/Servers

The multipart Filtered Cost Map resource and the multipart Endpoint Cost Service 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 incurring any incompatibility problem.

9.2. Compatibility with Multi-Cost Extension

The extension defined in this document is compatible with the multi-cost extension [RFC8189]. Such a resource has a media type of either "multipart/related; type=application/alto-costmap+json" or "multipart/related; type=application/alto-endpointcost+json". Its "cost-constraints" field must either be "false" or not present and the Path Vector cost type must be present in the "cost-type-names" capability field but must not be present in the "testable-cost-type-names" field, as specified in Section 7.2.4 and Section 7.3.4.

9.3. Compatibility with Incremental Update

This extension is compatible with the incremental update extension [RFC8895]. ALTO clients and servers MUST follow the specifications given in Sections 5.2 and 6.7.3 of [RFC8895] to support incremental updates for a Path Vector resource.

9.4. Compatibility with Cost Calendar

The extension specified in this document is compatible with the Cost Calendar extension [RFC8896]. 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, a property of a single ANE may also have different values in different time intervals. In this case, if such an ANE has different property values in two time intervals, it MUST be treated as two different ANEs, i.e., with different entity identifiers. However, if it has the same property values in two time intervals, it MAY use the same identifier.

This rule allows the Path Vector extension to represent both changes of ANEs and changes of the ANEs' properties in a uniform way. The Path Vector part is calendared in a compatible way, and the Property Map part is not affected by the calendar extension.

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 bespoke language for general-purpose boolean tests can be a complex undertaking, and it is conceivable that there are some existing implementations already (the authors have not done an exhaustive search to determine whether there are such implementations). One avenue to develop such a language may be to explore extending current query languages like XQuery [XQuery] or JSONiq [JSONiq] and integrating these with ALTO.

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

10.2. General Multi-Resource Query

Querying multiple ALTO information resources continuously is a general requirement. Enabling such a capability, however, must address general issues like efficiency and consistency. The incremental update extension [RFC8895] supports submitting multiple queries in a single request, and allows flexible control over the queries. However, it does not cover the case introduced in this document where multiple resources are needed for a single request.

This extension gives an example of using a multipart message to encode the responses from two specific ALTO information resources: a Filtered Cost Map or an Endpoint Cost Service, and a Property Map. By packing multiple resources in a single response, the implication is that servers may proactively push related information resources to clients.

Thus, it is worth looking into the direction of extending the SSE mechanism as used in the incremental update extension [RFC8895], or upgrading to HTTP/2 [I-D.ietf-httpbis-http2bis] and HTTP/3 [I-D.ietf-quic-http], which provides the ability to multiplex queries and to allow servers proactively send related information resources.

Defining a general multi-resource query mechanism is out of the scope of this document.

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 scrutiny on three security considerations discussed in the base protocol: confidentiality of ALTO information (Section 15.3 of [RFC7285]), potential undesirable guidance from authenticated ALTO information (Section 15.2 of [RFC7285]), and availability of ALTO service (Section 15.5 of [RFC7285]).

For confidentiality of ALTO information, a network operator should be aware that this extension may introduce a new risk: the Path Vector information, when used together with sensitive ANE properties such as capacities of bottleneck links, 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 attacker may identify the bottleneck link and start a distributed denial-of-service (DDoS) attack involving minimal flows to conduct the in-network congestion. Given the potential risk of leaking sensitive information, the Path Vector extension is mainly applicable in scenarios where 1) the ANE structures and ANE properties do not impose security risks to the ALTO service provider, e.g., not carrying sensitive information, or 2) the ALTO server and client have established a reliable trust relationship, for example, operated in the same administrative domain, or managed by business partners with legal contracts.

Three risk types are identified in Section 15.3.1 of [RFC7285]: (1) Excess disclosure of the ALTO service provider's data to an unauthorized ALTO client; (2) Disclosure of the ALTO service provider's data (e.g., network topology information or endpoint addresses) to an unauthorized third party; and (3) Excess retrieval of the ALTO service provider's data by collaborating ALTO clients. To mitigate these risks, an ALTO server MUST follow the guidelines in Section 15.3.2 of [RFC7285]. Furthermore, an ALTO server MUST follow the following additional protections strategies for risk types (1) and (3).

For risk type (1), an ALTO server MUST use the authentication methods specified in Section 15.3.2 of [RFC7285] to authenticate the identify of an ALTO client, and apply access control techniques to restrict unprivileged ALTO clients from retrieving sensitive Path Vector information. For settings where the ALTO server and client are not in the same trust domain, the ALTO server should reach agreements with the ALTO client on protecting the confidentiality before granting the access to Path Vector service with sensitive information. Such agreements may include legal contracts or Digital Right Management (DRM) techniques. Otherwise, the ALTO server MUST NOT offer the Path Vector service carrying sensitive information to the clients unless the potential risks are fully assessed and mitigated.

For risk type (3), an ALTO service provider must be aware that persistent ANEs may be used as "landmarks" in collaborative inferences. Thus, they should only be used when exposing public service access points (e.g., API gateways, CDNi) and/or when the granularity is coarse-grained (e.g., when an ANE represents an AS, a data center or a WAN). Otherwise, an ALTO server MUST use dynamic mappings from ephemeral ANE names to underlying physical entities. Specifically, for the same physical entity, an ALTO server SHOULD assign a different ephemeral ANE name when the entity appears in the responses to different clients or even for different request from the same client. A RECOMMENDED assignment strategy is to generate ANE names from random numbers.

Further, to protect the network topology from graph reconstruction (e.g., through isomorphic graph identification [BONDY]), the ALTO server SHOULD consider protection mechanisms to reduce information exposure or obfuscate the real information. When doing so, the ALTO server must be aware that information reduction/obfuscation may lead to potential Undesirable Guidance from Authenticated ALTO Information risk (Section 15.2 of [RFC7285]).

Thus, implementations of ALTO servers involving reduction or obfuscation of the Path Vector information SHOULD consider reduction/obfuscation mechanisms that can preserve the integrity of ALTO information, for example, by using minimal feasible region compression algorithms [NOVA] or obfuscation protocols [RESA][MERCATOR]. However, these obfuscation methods are experimental and their practical applicability of these methods to the generic capability provided by this extension is not fully assessed. The ALTO server MUST carefully verify that the deployment scenario satisfies the security assumptions of these methods before applying them to protect Path Vector services with sensitive network information.

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 consume intolerable amounts of the server-side computation and storage, which can break the ALTO server. For example, if an ALTO server implementation dynamically computes the Path Vectors for each request, the service providing Path Vectors may become an entry point for denial-of-service attacks on the availability of an ALTO server.

To mitigate this risk, an ALTO server may consider using optimizations such as precomputation-and-projection mechanisms [MERCATOR] to reduce the overhead for processing each query. Also, an ALTO server may also protect itself from malicious clients by monitoring the behaviors of clients and stopping serving clients with suspicious behaviors (e.g., sending requests at a high frequency).

The ALTO service providers must be aware that providing incremental updates of the "max-reservable-bandwidth" may provide information about other consumers of the network. For example, a change of the value may indicate one or more reservations has been made or changed. To mitigate this risk, an ALTO server can batch the updates and/or add a random delay before publishing the updates.

12. IANA Considerations

12.1. ALTO Cost Metric Registry

This document registers a new entry to the ALTO Cost Metric Registry, as instructed by Section 14.2 of [RFC7285]. The new entry is as shown below in Table 1.

| Identifier | Intended Semantics | Security Considerations |
|------------|--------------------|-------------------------|
| ane-path | See Section 6.5.1 | See Section 11 |

Table 1: ALTO Cost Metric Registry

12.2. ALTO Cost Mode Registry

This document registers a new entry to the ALTO Cost Mode Registry, as instructed by Section 4 of [I-D.bw-alto-cost-mode]. The new entry is as shown below in Table 2.

| Identifier | Intended Semantics |
|------------|--------------------|
| array | See Section 6.5.2 |

Table 2: ALTO Cost Mode Registry

12.3. ALTO Entity Domain Type Registry

This document registers a new entry to the ALTO Domain Entity Type Registry, as instructed by Section 12.2 of [I-D.ietf-alto-unified-props-new]. The new entry is as shown below in Table 3.

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

Table 3: ALTO Entity Domain Type Registry

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.

Mapping to ALTO Address Type: This entity type does not map to ALTO address type.

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.

12.4. ALTO Entity Property Type Registry

Two initial entries "max-reservable-bandwidth" and "persistent-entity-id" 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 4 and their details can be found in Section 12.4.1 and Section 12.4.2.

| Identifier | Intended Semantics | Media Type of Defining Resource |
|--------------------------|--------------------|---------------------------------|
| max-reservable-bandwidth | See Section 6.4.1 | application/alto-propmap+json |
| persistent-entity-id | See Section 6.4.2 | application/alto-propmap+json |

Table 4: Initial Entries for ane Domain in the ALTO Entity Property Types Registry

12.4.1. New ANE Property Type: Maximum Reservable Bandwidth

Identifier: "max-reservable-bandwidth"

Intended Semantics: See Section 6.4.1.

Media Type of Defining Resource: application/alto-propmap+json

Security Considerations: This property is essential for applications such as large-scale data transfers or overlay network interconnection to make better choice of bandwidth reservation. It may reveal the bandwidth usage of the underlying network and can potentially be leveraged to reduce the cost of conducting

denial-of-service attacks. Thus, the ALTO server MUST consider protection mechanisms including only providing the information to authorized clients, and information reduction and obfuscation as introduced in Section 11.

12.4.2. New ANE Property Type: Persistent Entity ID

Identifier: "persistent-entity-id"

Intended Semantics: See Section 6.4.2.

Media Type of Defining Resource: application/alto-propmap+json

Security Considerations: This property is useful when an ALTO server wants to selectively expose certain service points whose detailed properties can be further queried by applications. The entity IDs may consider sensitive information about the underlying network, and an ALTO server should follow the security considerations in Section 11 of [I-D.ietf-alto-unified-props-new].

13. References

13.1. Normative References

[I-D.bw-alto-cost-mode]

Boucadair, M. and Q. Wu, "A Cost Mode Registry for the Application-Layer Traffic Optimization (ALTO) Protocol", Work in Progress, Internet-Draft, draft-bw-alto-cost-mode-01, 4 March 2022, <<https://datatracker.ietf.org/doc/html/draft-bw-alto-cost-mode-01>>.

[I-D.ietf-alto-unified-props-new]

Roome, W., Randriamasy, S., Yang, Y. R., Zhang, J. J., and K. Gao, "An ALTO Extension: Entity Property Maps", Work in Progress, Internet-Draft, draft-ietf-alto-unified-props-new-24, 28 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-unified-props-new-24>>.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/rfc/rfc2046>>.

[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/rfc/rfc2119>>.

- [RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, DOI 10.17487/RFC2387, August 1998, <<https://www.rfc-editor.org/rfc/rfc2387>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/rfc/rfc5322>>.
- [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/rfc/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/rfc/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/rfc/rfc8189>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/rfc/rfc8895>>.
- [RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <<https://www.rfc-editor.org/rfc/rfc8896>>.

13.2. Informative References

- [BONDY] Bondy, J.A. and R.L. Hemminger, "Graph reconstructiona survey", Journal of Graph Theory, Volume 1, Issue 3, pp 227-268 , 1977, <<https://doi.org/10.1002/jgt.3190010306>>.
- [BOXOPT] 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, <<https://doi.org/10.1609/aaai.v33i01.33011674>>.

- [CLARINET] Viswanathan, R., Ananthanarayanan, G., and A. Akella, "CLARINET: WAN-Aware Optimization for Analytics Queries", In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association, Savannah, GA, 435-450 , 2016, <<https://dl.acm.org/doi/abs/10.5555/3026877.3026911>>.
- [G2] Ros-Giralt, J., Bohara, A., Yellamraju, S., Langston, M.H., Lethin, R., Jiang, Y., Tassiulas, L., Li, J., Tan, Y., and M. Veeraraghavan, "On the Bottleneck Structure of Congestion-Controlled Networks", Proceedings of the ACM on Measurement and Analysis of Computing Systems, Volume 3, Issue 3, pp 1-31 , 2019, <<https://dl.acm.org/doi/10.1145/3366707>>.
- [HUG] Chowdhury, M., Liu, Z., Ghodsi, A., and I. Stoica, "HUG: Multi-Resource Fairness for Correlated and Elastic Demands", 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16) (Santa Clara, CA, 2016), 407-424. , 2016, <<https://dl.acm.org/doi/10.5555/2930611.2930638>>.
- [I-D.ietf-alto-performance-metrics]
Wu, Q., Yang, Y. R., Lee, Y., Dhody, D., Randriamasy, S., and L. M. C. Murillo, "ALTO Performance Cost Metrics", Work in Progress, Internet-Draft, draft-ietf-alto-performance-metrics-26, 2 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-performance-metrics-26>>.
- [I-D.ietf-httpbis-http2bis]
Thomson, M. and C. Benfield, "HTTP/2", Work in Progress, Internet-Draft, draft-ietf-httpbis-http2bis-07, 24 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-http2bis-07>>.
- [I-D.ietf-quic-http]
Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-34, 2 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>>.
- [JSONiq] "The JSON Query language", 2020, <<https://www.jsoniq.org/>>.

- [MERCATOR] Xiang, Q., Zhang, J., Wang, X., Liu, Y., Guok, C., Le, F., MacAuley, J., Newman, H., and Y.R. Yang, "Toward Fine-Grained, Privacy-Preserving, Efficient Multi-Domain Network Resource Discovery", IEEE/ACM IEEE Journal on Selected Areas of Communication 37(8): 1924-1940, 2019, <<https://doi.org/10.1109/JSAC.2019.2927073>>.
- [MOWIE] Zhang, Y., Li, G., Xiong, C., Lei, Y., Huang, W., Han, Y., Walid, A., Yang, Y.R., and Z. Zhang, "MoWIE: Toward Systematic, Adaptive Network Information Exposure as an Enabling Technique for Cloud-Based Applications over 5G and Beyond", In Proceedings of the Workshop on Network Application Integration/CoDesign, ACM, Virtual Event USA, 20-27. , 2020, <<https://doi.org/10.1145/3405672.3409489>>.
- [NOVA] 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, <<https://doi.org/10.1109/IWQoS.2017.7969117>>.
- [RESA] Xiang, Q., Zhang, J., Wang, X., Liu, Y., Guok, C., Le, F., MacAuley, J., Newman, H., and Y.R. Yang, "Fine-grained, multi-domain network resource abstraction as a fundamental primitive to enable high-performance, collaborative data sciences", Proceedings of the Super Computing 2018, 5:1-5:13 , 2019, <<https://doi.org/10.1109/SC.2018.000008>>.
- [RFC2216] Shenker, S. and J. Wroclawski, "Network Element Service Specification Template", RFC 2216, DOI 10.17487/RFC2216, September 1997, <<https://www.rfc-editor.org/rfc/rfc2216>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/rfc/rfc4271>>.
- [SENSE] "Software Defined Networking (SDN) for End-to-End Networked Science at the Exascale", 2019, <<https://www.es.net/network-r-and-d/sense/>>.
- [SEREDGE] Contreras, L., Baliosian, J., Martnez-Julia, P., and J. Serrat, "Computing at the Edge: But, what Edge?", In proceedings of the NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium. pp. 1-9. , 2020, <<https://doi.org/10.1109/NOMS47738.2020.9110342>>.

- [SWAN] Hong, C., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., and R. Wattenhofer, "Achieving High Utilization with Software-driven WAN", In Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM '13), ACM, New York, NY, USA, 15-26. , 2013, <<http://doi.acm.org/10.1145/2486001.2486012>>.
- [UNICORN] Xiang, Q., Chen, S., Gao, K., Newman, H., Taylor, I., Zhang, J., and Y.R. Yang, "Unicorn: Unified Resource Orchestration for Multi-Domain, Geo-Distributed Data Analytics", 2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) (Aug. 2017), 1-6. , 2017, <<https://doi.org/10.1016/j.future.2018.09.048>>.
- [XQuery] "XQuery 3.1: An XML Query Language", 2017, <<https://www.w3.org/TR/xquery-31/>>.

Appendix A. 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, Dawn Chen, Wendy Roome, and Michael Scharf for their contributions to earlier drafts.

The authors would also like to thank Tim Chown, Luis Contreras, Roman Danyliw, Benjamin Kaduk, Erik Kline, Suresh Krishnan, Murray Kucherawy, Warren Kumari, Danny Lachos, Francesca Palombini, Eric Vyncke, Samuel Weiler, and Qiao Xiang whose feedback and suggestions are invaluable to improve the practicability and conciseness of this document, and Mohamed Boucadair, Martin Duke, Vijay Gurbani, Jan Seedorf, and Qin Wu who provide great support and guidance.

Appendix B. Revision Logs (To be removed before publication)

B.1. Changes since -20

Reivision -21

- * changes the normative requirement on protecting confidentiality of PV information with softer language

B.2. Changes since -19

Revision -20

- * changes the IANA registry information
- * adopts the comments from IESG reviews

B.3. Changes since -18

Revision -19

- * adds detailed examples for use cases
- * clarify terms with ambiguous meanings

B.4. Changes since -17

Revision -18

- * changes the specification for content-id to conform to [RFC2387] and [RFC5322]
- * adds IPv6 examples

B.5. Changes since -16

Revision -17

- * adds items for media type of defining resources in IANA considerations

B.6. Changes since -15

Revision -16

- * resolves the compatibility with the Multi-Cost extension (RFC 8189)
- * adds media types of defining resources for ANE property types (for IANA registration)

B.7. Changes since -14

Revision -15

- * fixes the IDNits warnings,

- * fixes grammar issues,
- * addresses the comments in the AD review.

B.8. Changes since -13

Revision -14

- * addresses the comments in the chair review,
- * fixes most issues raised by IDNits.

B.9. Changes since -12

Revision -13

- * changes the abstract based on the chairs' reviews
- * integrates Richard's responds to WGLC reviews

B.10. Changes since -11

Revision -12

- * clarifies the definition of ANEs in a similar way as how Network Elements is defined in [RFC2216]
- * restructures several paragraphs that are not clear (Sec 3, Path Vector bullet, Sec 4.2, Sec 5.1.3, Sec 6.2.4, Sec 6.4.2, Sec 9.3)
- * uses "ALTO Entity Domain Type Registry"

B.11. 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.

B.12. 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

B.13. 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

B.14. 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
Sichuan University
No.24 South Section 1, Yihuan Road
Chengdu
610000
China
Email: kaigao@scu.edu.cn

Young Lee
Samsung
South Korea
Email: younglee.tx@gmail.com

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
Tongji University
4800 Caoan Road
Shanghai
201804
China
Email: jingxuan.n.zhang@gmail.com

ALTO Working Group
Internet-Draft
Intended status: Standards Track
Expires: 22 September 2022

Q. Wu
Huawei
Y. Yang
Yale University
Y. Lee
Samsung
D. Dhody
Huawei
S. Randriamasy
Nokia Bell Labs
L. Contreras
Telefonica
21 March 2022

ALTO Performance Cost Metrics
draft-ietf-alto-performance-metrics-28

Abstract

The cost metric is a basic concept in Application-Layer Traffic Optimization (ALTO), and different applications may use different types of cost metrics. Since the ALTO base protocol (RFC 7285) defines only a single cost metric (namely, the generic "routingcost" metric), if an application wants to issue a cost map or an endpoint cost request in order to identify a resource provider that offers better performance metrics (e.g., lower delay or loss rate), the base protocol does not define the cost metric to be used.

This document addresses this issue by extending the specification to provide a variety of network performance metrics, including network delay, delay variation (a.k.a, jitter), packet loss rate, hop count, and bandwidth.

There are multiple sources (e.g., estimation based on measurements or service-level agreement) to derive a performance metric. This document introduces an additional "cost-context" field to the ALTO "cost-type" field to convey the source of a performance metric.

Status of This Memo

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

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

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

This Internet-Draft will expire on 22 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 3 |
| 2. Requirements Language | 6 |
| 3. Performance Metric Attributes | 6 |
| 3.1. Performance Metric Context: "cost-context" | 7 |
| 3.2. Performance Metric Statistics | 9 |
| 4. Packet Performance Metrics | 11 |
| 4.1. Cost Metric: One-Way Delay (delay-ow) | 11 |
| 4.1.1. Base Identifier | 11 |
| 4.1.2. Value Representation | 12 |
| 4.1.3. Intended Semantics and Use | 12 |
| 4.1.4. Cost-Context Specification Considerations | 14 |
| 4.2. Cost Metric: Round-trip Delay (delay-rt) | 16 |
| 4.2.1. Base Identifier | 16 |
| 4.2.2. Value Representation | 16 |
| 4.2.3. Intended Semantics and Use | 16 |
| 4.2.4. Cost-Context Specification Considerations | 17 |
| 4.3. Cost Metric: Delay Variation (delay-variation) | 18 |
| 4.3.1. Base Identifier | 18 |
| 4.3.2. Value Representation | 18 |
| 4.3.3. Intended Semantics and Use | 18 |
| 4.3.4. Cost-Context Specification Considerations | 19 |
| 4.4. Cost Metric: Loss Rate (lossrate) | 20 |
| 4.4.1. Base Identifier | 20 |
| 4.4.2. Value Representation | 20 |
| 4.4.3. Intended Semantics and Use | 20 |

| | | |
|--------|---|----|
| 4.4.4. | Cost-Context Specification Considerations | 21 |
| 4.5. | Cost Metric: Hop Count (hopcount) | 22 |
| 4.5.1. | Base Identifier | 22 |
| 4.5.2. | Value Representation | 22 |
| 4.5.3. | Intended Semantics and Use | 22 |
| 4.5.4. | Cost-Context Specification Considerations | 23 |
| 5. | Throughput/Bandwidth Performance Metrics | 24 |
| 5.1. | Cost Metric: TCP Throughput (tput) | 24 |
| 5.1.1. | Base Identifier | 24 |
| 5.1.2. | Value Representation | 24 |
| 5.1.3. | Intended Semantics and Use | 24 |
| 5.1.4. | Cost-Context Specification Considerations | 25 |
| 5.2. | Cost Metric: Residual Bandwidth (bw-residual) | 26 |
| 5.2.1. | Base Identifier | 26 |
| 5.2.2. | Value Representation | 26 |
| 5.2.3. | Intended Semantics and Use | 26 |
| 5.2.4. | Cost-Context Specification Considerations | 28 |
| 5.3. | Cost Metric: Available Bandwidth (bw-available) | 28 |
| 5.3.1. | Base Identifier | 28 |
| 5.3.2. | Value Representation | 28 |
| 5.3.3. | Intended Semantics and Use | 29 |
| 5.3.4. | Cost-Context Specification Considerations | 30 |
| 6. | Operational Considerations | 30 |
| 6.1. | Source Considerations | 31 |
| 6.2. | Metric Timestamp Consideration | 31 |
| 6.3. | Backward Compatibility Considerations | 31 |
| 6.4. | Computation Considerations | 32 |
| 6.4.1. | Configuration Parameters Considerations | 32 |
| 6.4.2. | Aggregation Computation Considerations | 32 |
| 7. | Security Considerations | 32 |
| 8. | IANA Considerations | 33 |
| 9. | Acknowledgments | 35 |
| 10. | References | 35 |
| 10.1. | Normative References | 35 |
| 10.2. | Informative References | 37 |
| | Authors' Addresses | 38 |

1. Introduction

Application-Layer Traffic Optimization (ALTO) provides a means for network applications to obtain network information so that the applications can identify efficient application-layer traffic patterns using the networks. Cost metrics are 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. However, the ALTO base protocol has registered only a single cost metric, i.e., the generic "routingcost" metric (Section 14.2 of [RFC7285]); no latency or bandwidth related metrics are defined in the base protocol.

This document registers a set of new cost metrics (Table 1) to allow applications to determine "where" to connect based on network performance criteria including delay and bandwidth related metrics.

| Metric | Definition in this doc | Semantics Based On |
|---------------------|------------------------|---|
| One-way Delay | Section 4.1 | Base: [RFC7471,8570,8571] sum Unidirectional Delay |
| Round-trip Delay | Section 4.2 | Base: Sum of two directions from above |
| Delay Variation | Section 4.3 | Base: [RFC7471,8570,8571] sum of Unidirectional Delay Variation |
| Loss Rate | Section 4.4 | Base: [RFC7471,8570,8571] aggr Unidirectional Link Loss |
| Residual Bandwidth | Section 5.2 | Base: [RFC7471,8570,8571] min Unidirectional Residual BW |
| Available Bandwidth | Section 5.3 | Base: [RFC7471,8570,8571] min Unidirectional Avail. BW |
| TCP Throughput | Section 5.1 | [I-D.ietf-tcpm-rfc8312bis] |
| Hop Count | Section 4.5 | [RFC7285] |

Table 1. Cost Metrics Defined in this Document.

The first 6 metrics listed in Table 1 (i.e., One-way Delay, Round-trip Delay, Delay Variation, Loss Rate, Residual Bandwidth, and Available Bandwidth) are derived from the set of traffic engineering performance metrics commonly defined in OSPF [RFC3630], [RFC7471]; IS-IS [RFC5305], [RFC8570]; and BGP-LS [RFC8571]. Deriving ALTO cost performance metrics from existing network-layer traffic engineering performance metrics, to expose to application-layer traffic optimization, can be a typical mechanism by network operators to deploy ALTO [RFC7971], [FlowDirector]. This document defines the base semantics of these metrics by extending them from link metrics

to end-to-end metrics for ALTO. The "Semantics Based On" column specifies at a high level how the end-to-end metric is computed from link metrics; the details will be specified in the following sections.

The common metrics Min/Max Unidirectional Delay defined in [RFC8570][RFC8571] and Max Link Bandwidth defined in [RFC3630,RFC5305] are not listed in Table 1 because they can be handled by applying the statistical operators defined in this document. The metrics related with utilized bandwidth and reservable bandwidth (i.e., Max Reservable BW and Unreserved BW defined in [RFC3630,RFC5305]) are outside the scope of this document.

The 7th metric (the estimated TCP-flow throughput metric) provides an estimation of the bandwidth of a TCP flow, using TCP throughput modeling, to support use cases of adaptive applications [Prophet], [G2]. Note that other transport-specific metrics can be defined in the future. For example, QUIC-related metrics [RFC9000] can be considered when the methodology to measure such metrics is more mature (e.g., [I-D.corre-quick-throughput-testing]).

The 8th metric (the hop count metric) in Table 1 is mentioned in the ALTO base protocol [RFC7285], but not defined, and this document defines it to be complete.

These 8 performance metrics can be classified into two categories: those derived from the performance of individual packets (i.e., One-way Delay, Round-trip Delay, Delay Variation, Loss Rate, and Hop Count), and those related to bandwidth/throughput (Residual bandwidth, and Available Bandwidth, and TCP throughput). These two categories are defined in Sections 4 and 5 respectively. Note that all metrics except Round-trip Delay are unidirectional. An ALTO client will need to query both directions if needed.

The purpose of this document is to ensure proper usage of these 8 performance metrics in the context of ALTO. 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 specified in Table 1.

The definitions of the intended semantics of the metrics tend to be coarse-grained, for guidance only, and they may work well for ALTO. On the other hand, a performance measurement framework, such as the IP Performance Measurement (IPPM) framework, may provide more details in defining a performance metric. This document introduces a mechanism called "cost-context" to provide additional details, when they are available; see Section 3.

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. In particular, [RFC7285] specifies that cost values should be assumed by default as JSONNumber. When defining the value representation of each metric in Table 1, this document conforms to [RFC7285], but specifies additional, generic constraints on valid JSONNumbers for each metric. For example, each new metric in Table 1 will be specified as non-negative (≥ 0); Hop Count is specified to be an integer.

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; 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 announces the metric in its information resource directory (IRD) as defined in Section 9.2 of [RFC7285].

An ALTO server introducing these metrics should consider related 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. Section 7 discusses security considerations in more detail.

2. Requirements Language

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.

3. Performance Metric Attributes

The definitions of the metrics in this document are coarse-grained, based on network-layer traffic engineering performance metrics, for guidance only. A fine-grained framework specified in [RFC6390] requires that the fine-grained specification of a network performance metric include 6 components: (i) Metric Name, (ii) Metric Description, (iii) Method of Measurement or Calculation, (iv) Units of Measurement, (v) Measurement Points, and (vi) Measurement Timing. Requiring that an ALTO server provides precise, fine-grained values for all 6 components for each metric that it exposes may not be feasible or necessary for all ALTO use cases. For example, an ALTO server computing its metrics from network-layer traffic-engineering

performance metrics may not have information about the method of measurement or calculation (e.g., measured traffic patterns).

To address the issue and realize ALTO use cases, for metrics in Table 1, this document defines performance metric identifiers which can be used in the ALTO protocol with well-defined (i) Metric Name, (ii) Metric Description, (iv) Units of Measurement, and (v) Measurement Points, which are always specified by the specific ALTO services; for example, endpoint cost service is between the two endpoints. Hence, the ALTO performance metric identifiers provide basic metric attributes.

To allow the flexibility of allowing an ALTO server to provide fine-grained information such as Method of Measurement or Calculation, according to its policy and use cases, this document introduces context information so that the server can provide these additional details.

3.1. Performance Metric Context: "cost-context"

The core additional details of a performance metric specify "how" the metric is obtained. This is referred to as the source of the metric. Specifically, this document defines three types of coarse-grained metric information sources: "nominal", and "sla" (service level agreement), and "estimation".

For a given type of source, precise interpretation of a performance metric value can depend on specific measurement and computation parameters.

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;
```

"cost-context" will not be used as a key to distinguish among performance metrics. Hence, an ALTO information resource MUST NOT announce multiple CostType with the same "cost-metric", "cost-mode" and "cost-context". They must be placed into different information resources.

The "cost-source" field of the "cost-context" field is defined as a string consisting of only US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A, and U+0061-U+007A). The cost-source is used in this document to indicate a string of this format.

As mentioned above, this document defines three values for "cost-source": "nominal", "sla", and "estimation". The "cost-source" field of the "cost-context" field MUST be one registered in "ALTO Cost Source" registry (Section 8).

The "nominal" category indicates that the metric value 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 involved devices; latency typically does not have a nominal value.

The "sla" category indicates that the metric value 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 an "sla" metric, it is RECOMMENDED that the "parameters" field provide a link to the SLA definition.

The "estimation" category indicates that the metric value is computed through an estimation process. An ALTO server may compute "estimation" values by retrieving and/or aggregating information from routing protocols (e.g., [RFC7471], [RFC8570], [RFC8571]), traffic measurement management tools (e.g., TWAMP [RFC5357]), and measurement frameworks (e.g., IPPM), with corresponding operational issues. An illustration of potential information flows used for estimating these metrics is shown in Figure 1. Section 6 discusses in more detail the operational issues and how a network may address them.

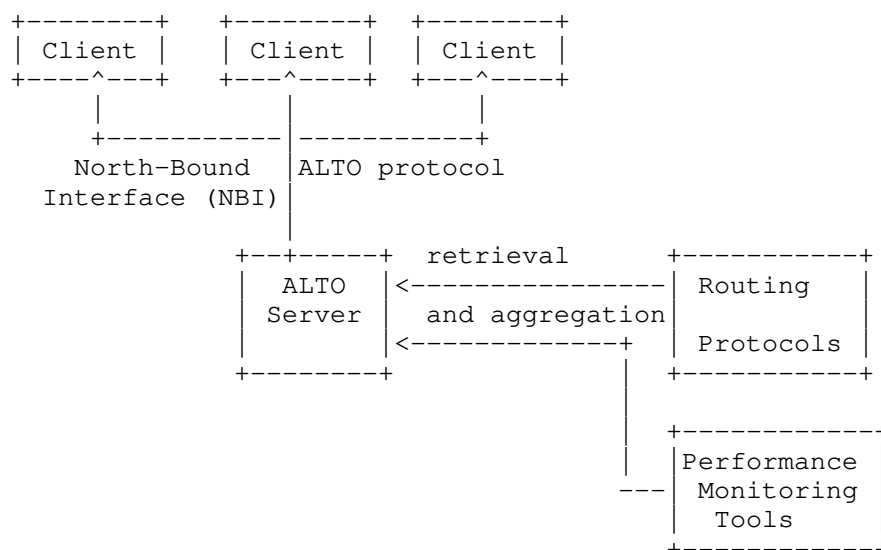


Figure 1. A framework to compute estimation to performance metrics

There can be multiple choices 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 source, i.e., "estimation".

3.2. 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. A statistical operator is applied to the samples to obtain a value to be reported to the client. Multiple statistical operators (e.g., min, median, and max) are commonly being used.

Hence, this document extends the general US-ASCII alphanumeric cost metric strings, formally specified as the CostMetric type defined in Section 10.6 of [RFC7285], as follows:

A cost metric string consists of a base metric identifier (or base identifier for short) string, followed by an optional statistical operator string, connected by the ASCII character colon (':', U+003A), if the statistical operator string exists. The total length of the cost metric string **MUST NOT** exceed 32, as required by [RFC7285].

The statistical operator string **MUST** be one of the following:

cur:

the instantaneous observation value of the metric from the most recent sample (i.e., the current value).

percentile, with letter 'p' followed by a number:

gives the percentile specified by the number following the letter 'p'. The number 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), followed by an optional decimal part, if a higher precision is needed. The decimal part should start with the '.' separator (U+002E), and followed by a sequence of one or more ASCII numbers between '0' and '9'. Assume this number is y and consider the samples coming from a random variable X. Then the metric returns x, such that the probability of X is less than or equal to x, i.e., $\text{Prob}(X \leq x) = y/100$. For example, delay-ow:p99 gives the 99% percentile of observed one-way delay; delay-ow:p99.9 gives the 99.9% percentile. Note that some systems use quantile, which is in the range [0, 1]. When there is a more common form for a given percentile, it is RECOMMENDED that the common form be used; that is, instead of p0, use min; instead of p50, use median; instead of p100, use max.

min:

the minimal value of the observations.

max:

the maximal value of the observations.

median:

the mid-point (i.e., p50) of the observations.

mean:

the arithmetic mean value of the observations.

stddev:

the standard deviation of the observations.

stdvar:

the standard variance of the observations.

Examples of cost metric strings then include "delay-ow", "delay-ow:min", "delay-ow:p99", where "delay-ow" is the base metric identifier string; "min" and "p99" are example statistical operator strings.

If a cost metric string does not have the optional statistical operator string, the statistical operator SHOULD be interpreted as the default statistical operator in the definition of the base metric. If the definition of the base metric does not provide a definition for the default statistical operator, the metric MUST be considered as the median value.

Note that RFC 7258 limits the overall cost metric identifier to 32 characters. The cost metric variants with statistical operator suffixes defined by this document are also subject to the same overall 32-character limit, so certain combinations of (long) base metric identifier and statistical operator will not be representable. If such a situation arises, it could be addressed by defining a new base metric identifier that is an "alias" of the desired base metric, with identical semantics and just a shorter name.

4. Packet Performance Metrics

This section introduces ALTO network performance metrics on one way delay, round-trip delay, delay variation, packet loss rate, and hop count. 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 when the packet enters the network to the time when the packet leaves the network (pkt.delay); whether the packet is dropped before reaching the destination (pkt.dropped); the number of network hops that the packet traverses (pkt.hopcount). The semantics of the performance metrics defined in this section are that they are statistics 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.

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

4.1.1. Base Identifier

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

4.1.2. Value Representation

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

4.1.3. Intended Semantics and Use

Intended Semantics: To specify the temporal and spatial aggregated delay of a stream of packets from the specified source to the specified destination. The base semantics of the metric is the Unidirectional Delay metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the delay for a link, it is the (temporal) aggregation of the link delays from the source to the destination. A non-normative reference definition of end-to-end one-way delay is [RFC7679]. The spatial aggregation level is specified in the query context, e.g., provider-defined identifier (PID) to PID, or endpoint to endpoint, where PID is defined in Section 5.1 of [RFC7285].

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: 239
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"
    ]
  }
}
```

HTTP/1.1 200 OK
Content-Length: 247
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
    }
  }
}
```

Note that since the "cost-type" does not include the "cost-source" field, the values are based on "estimation". Since the identifier does not include the statistical operator string component, the values will represent median values.

Example 1a below shows an example that is similar to Example 1, but for IPv6.

Example 1a: Delay value on source-destination endpoint pairs for IPv6

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 252
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": [
      "ipv6:2001:db8:100::1"
    ],
    "dsts": [
      "ipv6:2001:db8:100::2",
      "ipv6:2001:db8:100::3"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 257
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "delay-ow"
    }
  },
  "endpoint-cost-map": {
    "ipv6:2001:db8:100::1": {
      "ipv6:2001:db8:100::2": 10,
      "ipv6:2001:db8:100::3": 20
    }
  }
}
```

4.1.4. Cost-Context Specification Considerations

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

"sla": Many networks provide delay-related parameters in their application-level SLAs. It is RECOMMENDED that the "parameters" field of an "sla" one-way delay metric include a link (i.e., a field named "link") providing an URI to the specification of SLA details, if available. Such a specification can be either free text for possible presentation to the user, or a formal specification. The format of the specification is out of the scope of this document.

"estimation": The exact estimation method is out of the scope of this document. There can be multiple sources to estimate one-way delay. For example, the ALTO server may estimate the end-to-end delay by aggregation of routing protocol link metrics; the server may also estimate the delay using active, end-to-end measurements, for example, using the IPPM framework [RFC2330].

If the estimation is computed by aggregation of routing protocol link metrics (e.g., OSPF [RFC7471], IS-IS [RFC8570], or BGP-LS [RFC8571]) Unidirectional Delay link metrics, it is RECOMMENDED that the "parameters" field of an "estimation" one-way delay metric include the following information: (1) the RFC defining the routing protocol metrics (e.g., <https://www.rfc-editor.org/info/rfc7471> for RFC7471 derived metrics); (2) configurations of the routing link metrics such as configured intervals; and (3) the aggregation method from link metrics to end-to-end metrics. During aggregation from link metrics to the end-to-end metric, the server should be cognizant of potential issues when computing an end-to-end summary statistic from link statistics. The default end-to-end average one-way delay is the sum of average link one-way delays. If an ALTO server provides the min and max statistical operators for the one-way delay metric, the values can be computed directly from the routing link metrics, as [RFC7471,RFC8570,RFC8571] provide Min/Max Unidirectional Link Delay.

If the estimation is from the IPPM measurement framework, it is RECOMMENDED that the "parameters" field of an "estimation" one-way delay metric includes the following information: the URI to the URI field of the IPPM metric defined in the IPPM performance metric [IANA-IPPM] registry (e.g., https://www.iana.org/assignments/performance-metrics/OWDelay_Active_IP-UDP-Poisson-Payload250B_RFC8912sec7_Seconds_95Percentile). The IPPM metric MUST be one-way delay (i.e., IPPM OWDelay* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

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

4.2.1. Base Identifier

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

4.2.2. Value Representation

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

4.2.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial aggregated round-trip delay between the specified source and specified destination. The base semantics is that it is the sum of one-way delay from the source to the destination and the one-way delay from the destination back to the source, where the one-way delay is defined in Section 4.1. A non-normative reference definition of end-to-end round-trip delay is [RFC2681]. 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 (delay-ow) 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 of source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 238
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"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 245
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
    }
  }
}
```

4.2.4. Cost-Context Specification Considerations

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

"sla": See the "sla" entry in Section 4.1.4.

"estimation": See the "estimation" entry in Section 4.1.4. For estimation by aggregation of routing protocol link metrics, the aggregation should include all links from the source to the destination and then back to the source; for estimation using IPPM, the IPPM metric MUST be round-trip delay (i.e., IPPM RTDelay* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

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

4.3.1. Base Identifier

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

4.3.2. Value Representation

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

4.3.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial aggregated delay variation (also called delay jitter) with respect to the minimum delay observed on the stream over the one-way delay from the specified source and destination, where the one-way delay is defined in Section 4.1. A non-normative reference definition of end-to-end one-way delay variation is [RFC3393]. Note that [RFC3393] allows the specification of a generic selection function *F* to unambiguously define the two packets selected to compute delay variations. This document defines the specific case that *F* selects as the "first" packet the one with the smallest one-way delay. 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: 245
Content-Type: application/alto-endpointcostparams+json
Accept:
    application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "delay-variation"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 252
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "delay-variation"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 1
    }
  }
}
```

4.3.4. Cost-Context Specification Considerations

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

"sla": See the "sla" entry in Section 4.1.4.

"estimation": See the "estimation" entry in Section 4.1.4. For estimation by aggregation of routing protocol link metrics, the default aggregation of the average of delay variations is the sum of the link delay variations; for estimation using IPPM, the IPPM metric MUST be delay variation (i.e., IPPM OWPDV* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

4.4. Cost Metric: Loss Rate (lossrate)

4.4.1. Base Identifier

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

4.4.2. Value Representation

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

4.4.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial aggregated one-way packet loss rate from the specified source and the specified destination. The base semantics of the metric is the Unidirectional Link Loss metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the loss for a link, it is the aggregated loss of all links from the source to the 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: 238
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"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 248
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": 0.01
    }
  }
}
```

4.4.4. Cost-Context Specification Considerations

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

"sla": See the "sla" entry in Section 4.1.4..

"estimation": See the "estimation" entry in Section 4.1.4. For estimation by aggregation of routing protocol link metrics, the default aggregation of the average of loss rate is the sum of the link link loss rates. But this default aggregation is valid only if two conditions are met: (1) it is valid only when link loss rates are low, and (2) it assumes that each link's loss events are uncorrelated with every other link's loss events. When loss rates at the links are high but independent, the general formula for aggregating loss assuming each link is independent is to compute end-to-end loss as one minus the product of the success rate for each link. Aggregation when losses at links are correlated can be more complex and the ALTO server should be cognizant of correlated loss rates. For estimation using IPPM, the IPPM metric MUST be packet loss (i.e., IPPM OWLoss* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

4.5. Cost Metric: Hop Count (hopcount)

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

4.5.1. Base Identifier

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

4.5.2. Value Representation

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

4.5.3. 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 the number of router hops computed from the routing protocols originating this information. A hop, however, may represent other units. 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: 238
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"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 245
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
    }
  }
}
```

4.5.4. Cost-Context Specification Considerations

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

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

"estimation": The exact estimation method is out of the scope of this document. An example of estimating hopcounts is by importing from IGP routing protocols. It is RECOMMENDED that the "parameters" field of an "estimation" hop count define the meaning of a hop.

5. Throughput/Bandwidth Performance Metrics

This section introduces four throughput/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.

5.1. Cost Metric: TCP Throughput (tput)

5.1.1. Base Identifier

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

5.1.2. Value Representation

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

5.1.3. Intended Semantics and Use

Intended Semantics: To give the throughput of a TCP congestion-control conforming flow from the specified source to the specified destination. The throughput SHOULD be interpreted as only an estimation, and the estimation is designed only for bulk flows.

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: 234
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"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 251
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
    }
  }
}
```

5.1.4. Cost-Context Specification Considerations

"nominal": Typically TCP throughput does not have a nominal value, and SHOULD NOT be generated.

"sla": Typically TCP throughput does not have an SLA value, and SHOULD NOT be generated.

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" TCP throughput metric include the following information: (1) the congestion-control algorithm; and (2) the estimation methodology. To specify (1), it is RECOMMENDED that the "parameters" field (object) include a field named "congestion-control-algorithm", which provides a URI for the specification of the algorithm; for example, for an ALTO server to provide estimation to the throughput of a Cubic Congestion control flow, its "parameters" includes a field "congestion-control-algorithm", with value being set to [I-D.ietf-tcpm-rfc8312bis]; for an ongoing congestion control algorithm such as BBR, a link to its specification. To specify (2), the "parameters" includes as many details as possible; for example, for TCP Cubic throughput estimation, the "parameters" field specifies that the throughput is estimated by setting `_C_` to 0.4, and the Equation in Figure 8 of [I-D.ietf-tcpm-rfc8312bis] is applied; as an alternative, the methodology may be based on the NUM model [Prophet], or the G2 model [G2]. The exact specification of the parameters field is out of the scope of this document.

5.2. Cost Metric: Residual Bandwidth (bw-residual)

5.2.1. Base Identifier

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

5.2.2. Value Representation

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

5.2.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial residual bandwidth from the specified source and the specified destination. The base semantics of the metric is the Unidirectional Residual Bandwidth metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the residual bandwidth for a link, it is the residual bandwidth of the path from the source to the destination. Hence, it is the minimal residual bandwidth among all links from the source to the destination. When the max statistical operator is defined for the metric, it typically provides the minimum of the link capacities along the path, as the default value of the residual bandwidth of a link is its link capacity [RFC8571,8570,7471]. The spatial aggregation unit is specified in the query context (e.g., PID to PID,

or endpoint to endpoint).

The default statistical operator for residual bandwidth is the current instantaneous sample; that is, the default is assumed to be "cur".

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-residual value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
```

```
Host: alto.example.com
```

```
Content-Length: 241
```

```
Content-Type: application/alto-endpointcostparams+json
```

```
Accept:
```

```
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "bw-residual"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

HTTP/1.1 200 OK
Content-Length: 255
Content-Type: application/alto-endpointcost+json

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "bw-residual"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 2000
    }
  }
}
```

5.2.4. Cost-Context Specification Considerations

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

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

"estimation": See the "estimation" entry in Section 4.1.4 on aggregation of routing protocol link metrics. The current ("cur") residual bandwidth of a path is the minimal of the residual bandwidth of all links on the path.

5.3. Cost Metric: Available Bandwidth (bw-available)

5.3.1. Base Identifier

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

5.3.2. Value Representation

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

5.3.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial available bandwidth from the specified source to the specified destination. The base semantics of the metric is the Unidirectional Available Bandwidth metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the available bandwidth for a link, it is the available bandwidth of the path from the source to the destination. Hence, it is the minimal available bandwidth among all links from the source to the destination. The spatial aggregation unit is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

The default statistical operator for available bandwidth is the current instantaneous sample; that is, the default is assumed to be "cur".

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

Example 8: bw-available value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 244
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "bw-available"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

HTTP/1.1 200 OK
Content-Length: 255
Content-Type: application/alto-endpointcost+json

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "bw-available"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 2000
    }
  }
}
```

5.3.4. Cost-Context Specification Considerations

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

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

"estimation": See the "estimation" entry in Section 4.1.4 on aggregation of routing protocol link metrics. The current ("cur") available bandwidth of a path is the minimum of the available bandwidth of all links on the path.

6. 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 in the following sub-sections.

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, this document specifies common issues unless one metric has its unique challenges.

6.1. 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", "nominal" or "lsa", where "estimation" is a type of measurement data source, "nominal" is a type of static configuration, and "lsa" 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.

6.2. Metric Timestamp Consideration

Despite the introduction of the additional cost-context information, the metrics do not have a field to indicate the timestamps of the data used to compute the metrics. 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 [RFC8895], the client SHOULD assume that the metric is computed using a snapshot at the time that is approximated by the receiving time.

6.3. 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 "lsa". 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, whose "cost-source" is "lsa": an ALTO client that is not aware of this extension will ignore this field and consider the metric estimation.

To address the backward-compatibility issue, if a "cost-metric" is "routingcost" and the metric contains a "cost-context" field, then it MUST be "estimation"; if it is not, the client SHOULD reject the information as invalid.

6.4. 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.

6.4.1. Configuration Parameters Considerations

Performance metrics often depend on configuration parameters, and exposing such configuration parameters can help an ALTO client to better understand the exposed metrics. In particular, an ALTO server may be configured to compute a TE metric (e.g., packet loss rate) in fixed intervals, say every T seconds. To expose this information, the ALTO server may provide the client with two pieces of additional information: (1) when the metrics are last computed, and (2) when the metrics will be updated (i.e., the validity period of the exposed metric values). The ALTO server can expose these two pieces of information by using the HTTP response headers Last-Modified and Expires.

6.4.2. Aggregation Computation Considerations

An ALTO server may not be able to measure the performance metrics to be exposed. The basic issue is that the "source" information can often be link level. For example, routing protocols often measure and report only per link loss, not end-to-end loss; similarly, routing protocols report link level available bandwidth, not end-to-end available bandwidth. The ALTO server then needs to aggregate these data to provide an abstract and unified view that can be more useful to applications. The server should consider that different metrics may use different aggregation computation. For example, the end-to-end latency of a path is the sum of the latency of the links on the path; the end-to-end available bandwidth of a path is the minimum of the available bandwidth of the links on the path; in contrast, aggregating loss values is complicated by the potential for correlated loss events on different links in the path

7. 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, 15.2, and 15.3 of [RFC7285] remain of utmost importance. Indeed, Traffic Engineering (TE) performance is highly sensitive ISP information; therefore, sharing TE metric values in numerical mode requires full mutual confidence between the entities managing the ALTO server and the ALTO client. ALTO servers will most likely distribute numerical TE performance to ALTO 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.

To mitigate confidentiality risks during information transport of TE performance metrics, the operator should address the risk of ALTO information being leaked to malicious Clients or third parties, through attacks such as the person-in-the-middle (PITM) attacks. As specified in "Protection Strategies" (Section 15.3.2 of [RFC7285]), the ALTO Server should authenticate ALTO Clients when transmitting an ALTO information resource containing sensitive TE performance metrics. "Authentication and Encryption" (Section 8.3.5 of [RFC7285]) specifies that "ALTO Server implementations as well as ALTO Client implementations MUST support the "https" URI scheme of [RFC7230] and Transport Layer Security (TLS) of [RFC8446]".

8. 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 <<https://www.iana.org/assignments/alto-protocol/alto-protocol.xhtml#cost-metrics>>. This document requests to add the following entries to the "ALTO Cost Metric" registry.

| Identifier | Intended Semantics |
|-----------------|-------------------------|
| delay-ow | Section 4.1 of [RFCXXX] |
| delay-rt | Section 4.2 of [RFCXXX] |
| delay-variation | Section 4.3 of [RFCXXX] |
| lossrate | Section 4.4 of [RFCXXX] |
| hopcount | Section 4.5 of [RFCXXX] |
| tput | Section 5.1 of [RFCXXX] |
| bw-residual | Section 5.2 of [RFCXXX] |
| bw-available | Section 5.3 of [RFCXXX] |

* [Note to the RFC Editor]: Please replace RFCXXX with the RFC number assigned to this document.

This document requests the creation of the "ALTO Cost Source" registry. This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO cost source types. Second, it provides references to particular semantics of allocated cost source types to be applied by both ALTO servers and applications utilizing ALTO clients.

A new ALTO cost source can be added after IETF Review [RFC8126], to ensure that proper documentation regarding the new ALTO cost source and its security considerations have been provided. The RFC(s) documenting the new cost source should be detailed enough to provide guidance to both ALTO service providers and applications utilizing ALTO clients as to how values of the registered ALTO cost source should be interpreted. Updates and deletions of ALTO cost source follow the same procedure.

Registered ALTO address type identifiers MUST conform to the syntactical requirements specified in Section 3.1. Identifiers are to be recorded and displayed as strings.

Requests to add a new value to the registry MUST include the following information:

- * Identifier: The name of the desired ALTO cost source type.
- * Intended Semantics: ALTO cost source type 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 endpoint property should be interpreted.
- * Security Considerations: ALTO cost source types expose information to ALTO clients. ALTO service providers should be made aware of the security ramifications related to the exposure of a cost source type.

This specification requests registration of the identifiers "nominal", "sla", and "estimation" listed in the table below. Semantics for these are documented in Section 3.1, and security considerations are documented in Section 7.

| Identifier | Intended Semantics | Security Considerations |
|------------|---|-------------------------|
| nominal | Values in nominal cases; Section 3.1 of [RFCXXX] | Section 7 of [RFCXXX] |
| sla | Values reflecting service level agreement; Section 3.1 of [RFCXXXX] | Section 7 of [RFCXXX] |
| estimation | Values by estimation; Section 3.1 of [RFCXXX] | Section 7 of [RFCXXX] |

9. Acknowledgments

The authors of this document would also like to thank Martin Duke for the highly informative, thorough AD reviews and comments. We thank Christian Amsuess, Elwyn Davies, Haizhou Du, Kai Gao, Geng Li, Lili Liu, Danny Alex Lachos Perez, and Brian Trammell for the reviews and comments. We thank Benjamin Kaduk, Eric Kline, Francesca Palombini, Lars Eggert, Martin Vigoureux, Murraray Kucherawy, Roman Danyliw, Zaheduzzaman Sarker, Eric Vyncke for discussions and comments that improve this document.

10. References

10.1. Normative References

- [I-D.ietf-tcpm-rfc8312bis]
 Xu, L., Ha, S., Rhee, I., Goel, V., and L. Eggert, "CUBIC for Fast and Long-Distance Networks", Work in Progress, Internet-Draft, draft-ietf-tcpm-rfc8312bis-07, 4 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-tcpm-rfc8312bis-07.txt>>.
- [IANA-IPPM]
 IANA, "Performance Metrics Registry, <https://www.iana.org/assignments/performance-metrics/performance-metrics.xhtml>".
- [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>>.

- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [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>>.
- [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>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [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>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.
- [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>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.
- [RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <<https://www.rfc-editor.org/info/rfc8571>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/info/rfc8895>>.

10.2. Informative References

- [FlowDirector] Pujol, E., Poese, I., Zerwas, J., Smaragdakis, G., and A. Feldmann, "Steering Hyper-Giants' Traffic at Scale", ACM CoNEXT 2020, 2020.
- [G2] Ros-Giralt, J., Bohara, A., Yellamraju, S., and et. al., "On the Bottleneck Structure of Congestion-Controlled Networks", ACM SIGMETRICS 2019, 2020.
- [I-D.corre-quick-throughput-testing] Corre, K., "Framework for QUIC Throughput Testing", Work in Progress, Internet-Draft, draft-corre-quick-throughput-testing-00, 17 September 2021, <<https://www.ietf.org/archive/id/draft-corre-quick-throughput-testing-00.txt>>.
- [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.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.

- [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>>.
- [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>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [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>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.

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
United States of America
Email: yry@cs.yale.edu

Young Lee
Samsung
Email: young.lee@gmail.com

Dhruv Dhody
Huawei
Leela Palace
Bangalore 560008
Karnataka
India
Email: dhruv.ietf@gmail.com

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
91460 Nozay
France
Email: sabine.randriamasy@nokia-bell-labs.com

Luis Miguel Contreras Murillo
Telefonica
Madrid
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

ALTO WG
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

W. Roome
S. Randriamasy
Nokia Bell Labs
Y. Yang
Yale University
J. Zhang
Tongji University
July 8, 2019

Unified Properties for the ALTO Protocol
draft-ietf-alto-unified-props-new-08

Abstract

This document extends the Application-Layer Traffic Optimization (ALTO) Protocol [RFC7285] by generalizing the concept of "endpoint properties" to domains of other 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

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

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

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

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 4 |
| 2. Overview: Basic Concepts | 5 |
| 2.1. Entity | 5 |
| 2.2. Entity Property | 6 |
| 2.3. Property Map | 6 |
| 2.4. Information Resource | 6 |
| 2.5. Entity Domain | 7 |
| 2.5.1. Resource-Specific Entity Domain | 7 |
| 2.5.2. Relationship between Entity and Entity Domain | 7 |
| 2.5.3. Aggregated Entity Domain | 7 |
| 2.5.4. Resource-Specific Entity Property | 8 |
| 2.6. Scope of Property Map | 8 |
| 2.7. Entity Hierarchy and Property Inheritance | 9 |
| 3. Protocol Specification: Basic Data Type | 10 |
| 3.1. Entity Domain | 10 |
| 3.1.1. Entity Domain Type | 10 |
| 3.1.2. Entity Domain Name | 10 |
| 3.1.3. Entity Identifier | 11 |
| 3.1.4. Hierarchy and Inheritance | 12 |
| 3.2. Entity Property | 12 |
| 3.2.1. Entity Property Type | 12 |
| 3.2.2. Entity Property Name | 13 |
| 3.3. Information Resource Export | 13 |
| 3.3.1. Resource-Specific Entity Domain Export | 13 |
| 3.3.2. Entity Property Mapping Export | 14 |
| 4. Entity Domain Types | 14 |
| 4.1. Internet Address Domain Types | 14 |
| 4.1.1. IPv4 Domain | 14 |
| 4.1.2. IPv6 Domain | 15 |
| 4.1.3. Hierarchy and Inheritance of Internet Address Domains | 15 |
| 4.2. PID Domain | 16 |

| | | |
|---------|---|----|
| 4.2.1. | Entity Domain Type | 16 |
| 4.2.2. | Domain-Specific Entity Identifiers | 16 |
| 4.2.3. | Hierarchy and Inheritance | 17 |
| 4.2.4. | Relationship To Internet Addresses Domains | 17 |
| 4.3. | Internet Address Properties vs. PID Properties | 17 |
| 5. | Entity Domains and Property Mappings in Information Resources | 17 |
| 5.1. | Network Map Resource | 17 |
| 5.1.1. | Resource-Specific Entity Domain | 18 |
| 5.1.2. | Entity Property Mapping | 18 |
| 5.2. | Endpoint Property Resource | 18 |
| 5.2.1. | Resource-Specific Entity Domain | 18 |
| 5.2.2. | Entity Property Mapping | 19 |
| 5.3. | Property Map Resource | 19 |
| 6. | Property Map | 19 |
| 6.1. | Media Type | 19 |
| 6.2. | HTTP Method | 19 |
| 6.3. | Accept Input Parameters | 19 |
| 6.4. | Capabilities | 19 |
| 6.5. | Uses | 20 |
| 6.6. | Response | 20 |
| 7. | Filtered Property Map | 21 |
| 7.1. | Media Type | 21 |
| 7.2. | HTTP Method | 21 |
| 7.3. | Accept Input Parameters | 21 |
| 7.4. | Capabilities | 22 |
| 7.5. | Uses | 22 |
| 7.6. | Response | 22 |
| 8. | Impact on Legacy ALTO Servers and ALTO Clients | 24 |
| 8.1. | Impact on Endpoint Property Service | 24 |
| 8.2. | Impact on Resource-Specific Properties | 24 |
| 8.3. | Impact on the pid Property | 25 |
| 8.4. | Impact on Other Properties | 25 |
| 9. | Examples | 25 |
| 9.1. | Network Map | 25 |
| 9.2. | Property Definitions | 26 |
| 9.3. | Information Resource Directory (IRD) | 27 |
| 9.4. | Property Map Example | 29 |
| 9.5. | Filtered Property Map Example #1 | 30 |
| 9.6. | Filtered Property Map Example #2 | 31 |
| 9.7. | Filtered Property Map Example #3 | 32 |
| 9.8. | Filtered Property Map Example #4 | 33 |
| 10. | Security Considerations | 34 |
| 11. | IANA Considerations | 35 |
| 11.1. | application/alto-* Media Types | 35 |
| 11.2. | ALTO Entity Domain Type Registry | 36 |
| 11.2.1. | Consistency Procedure between ALTO Address Type Registry and ALTO Entity Domain Registry | 37 |
| 11.2.2. | ALTO Entity Domain Registration Process | 38 |

| | |
|--|----|
| 11.3. ALTO Entity Property Type Registry | 39 |
| 11.4. ALTO Resource-Specific Entity Domain Registries | 40 |
| 11.4.1. Network Map | 40 |
| 11.4.2. Endpoint Property | 40 |
| 11.5. ALTO Resource Entity Property Mapping Registries | 40 |
| 11.5.1. Network Map | 40 |
| 12. Acknowledgments | 41 |
| 13. Normative References | 41 |
| Authors' Addresses | 42 |

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 two limitations.

First, it allows properties to be associated with only a particular domain of entities, namely individual IP addresses. It is reasonable to think that collections of endpoints, as defined by CIDRs [RFC4632] or PIDs, may also have properties. Since the EPS cannot be extended to new entity domains, new services, with new request and response messages, would have to be defined for new entity domains.

Second, 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 processes costs lookups without querying the ALTO server. [RFC7285] does not define an equivalent 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. But 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.

This document specifies a new approach for defining and retrieving ALTO properties to address the two limitations. Specifically, this document addresses the first limitation by introducing a generic concept called ALTO Entity Domains, where an entity is a generalization of an endpoint to also represent, a PID, a network

element, or a cell in a cellular network, etc. As a consequence, ALTO Entity Domains defined in this document are a super-set of ALTO Address Types defined in [RFC7285]. Their exact relationship is specified in Section 11.2.1.

Entity domains and property names are extensible. New entity domains can be defined without revising the messages defined in this document, in the same way that new cost metrics and new endpoint properties can be defined without revising the messages defined in [RFC7285].

Additional, this document addresses the second limitation by defining two new types of resources, namely Property Map (see Section 6) and Filtered Property Map (see Section 7). The former is a GET-mode resource which returns the property values for all entities in a domain, and is analogous to a network map or a cost map in [RFC7285]. The latter is a POST-mode resource which 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.

This document subsumes the Endpoint Property Service defined in [RFC7285], although that service may be retained for legacy clients (see Section 8).

2. Overview: Basic Concepts

Before we define the specification of unified properties, there are several basic concepts which we need to introduce.

2.1. Entity

The entity concept generalizes the concept of the endpoint defined in Section 2.1 of [RFC7285]. An entity is an object that can be an endpoint and is identified 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 is even not related to any network address.

Examples of eligible entities are:

- o a PID, defined in [RFC7285], that has a provider defined human readable abstract identifier defined by a ALTO network map, which maps a PID to a set of ipv4 and ipv6 addresses;
- o an autonomous system (AS), that has an AS number (ASN) as its identifier and maps to a set of ipv4 and ipv6 addresses;
- o a region representing a country, that is identified by its country code defined by ISO 3166 and maps to a set of cellular addresses;

- o a TCP/IP network flow, that has a server defined identifier consisting of the defining TCP/IP 5-Tuple, , which is an example that all endpoints are entities while not all entities are endpoints;
- o a routing element, that is specified in [RFC7921] and includes routing capability information;
- o an abstract network element, that has a server defined identifier and represents a network node, link or their aggregation.

2.2. Entity Property

An entity property defines a property of an entity. It is similar to the endpoint property defined by Section 7.1 of [RFC7285], but can be general besides network-aware.

For example,

- o an "ipv4" entity may have a property whose value is an Autonomous System (AS) number indicating the AS which this IPv4 address is owned by;
- o a "pid" entity may have a property which indicates the central geographical location of endpoints included by it.

2.3. Property Map

An ALTO property map provides a set of properties for a set of entities. These entities may be in different types. For example, an ALTO property map may define the ASN property for both "ipv4" and "ipv6" entities.

2.4. Information Resource

This document uses the same definition of the information resource as defined by [RFC7285]. Each information resource usually has a JSON format representation following a specific schema defined by its media type.

For example, an ALTO network map resource is represented by a JSON object of type InfoResourceNetworkMap defined by the media type "application/alto-networkmap+json".

2.5. Entity Domain

An entity domain defines a set of entities in the same type. This type is also called the type of this entity domain.

Using entity domains, an ALTO property map can indicate which entities the ALTO client can query to get their properties.

2.5.1. Resource-Specific Entity Domain

To define an entity domain, one naive solution is to enumerate all entities in this entity domain. But it is inefficient when the size of the entity domain is large.

To avoid enumerating all entities, this document introduces an approach called "Resource-Specific Entity Domain" to define entity domains:

Each information resource may define several types of entity domains. And for each type of entity domain, an information resource can define at most one entity domain. For example, an ALTO network map resource can define an IPv4 domain, an IPv6 domain and a pid domain. In this document, these entity domains are called resource-specific entity domains. An ALTO property map only need to indicate which types of entity domain defined by which information resources can be queried, the ALTO client will know which entities are effective to be queried.

2.5.2. Relationship between Entity and Entity Domain

In this document, an entity is owned by exact one entity domain. It requires that when an ALTO client or server references an entity, it must indicate its entity domain explicitly. Even two entities in two different entity domains may reflect to the same physical or logical object, we treat them as different entities.

Because of this rule, although the resource-specific entity domain approach has no ambiguity, it may introduce redundancy.

2.5.3. Aggregated Entity Domain

Two entities in two different resource-specific entity domains may reflect to the same physical or logical object. For example, the IPv4 entity "192.0.2.34" in the IPv4 domain of the network map "netmap1" and the IPv4 entity "192.0.2.34" in the IPv4 domain of the network map "netmap2" should indicate the same Internet endpoint addressed by the IPv4 address "192.0.2.34".

Each entity in each resource-specific entity domain may only have part of properties of its associated physical or logical object. For example, the IPv4 entity in the IPv4 domain of the network map "netmap1" only has the PID property defined by "netmap1"; same to the IPv4 entity in the IPv4 domain of the network map "netmap2". If the ALTO client wants to get the complete properties, using the resource-specific entity domain, the ALTO client has to query the IPv4 entity "192.0.2.34" twice.

To simplify the query process of the ALTO client, this document introduces the concept "Aggregated Entity Domain". An aggregated entity domain defines an aggregated set of entities coming from multiple resource-specific entity domains in the same type. An entity in the aggregated entity domain includes all properties defined for the associated entity in each associated resource-specific entity domains. For example, the IPv4 entity "192.0.2.34" in the aggregated entity domain between the IPv4 domain of "netmap1" and the IPv4 domain of "netmap2" has PID properties defined by both "netmap1" and "netmap2".

2.5.4. Resource-Specific Entity Property

According to the example of the aggregated entity domain, an entity may have multiple properties in the same type but associated to different information resources. To distinguish them, this document uses the same approach proposed by Section 10.8.1 of [RFC7285], which is called "Resource-Specific Entity Property".

2.6. Scope of Property Map

Using entity domains to organize entities, an ALTO property map resource actually provides a set of properties for some entity domains. If we ignore the syntax sugar of the aggregated entity domain, we can consider an ALTO property map resource just provides a set of $(ri, di) \Rightarrow (ro, po)$ 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 $(ri, di) \Rightarrow (ro, po)$ mapping, the scope of an ALTO property map resource must be one of 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" points to the resulting property map resource, "r" presents 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 entities 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.

2.7. Entity Hierarchy and Property Inheritance

Enumerating all individual effective entities are inefficient. Some types of entities have the hierarchy format, e.g., `cidr`, which stand for sets of individual entities. Many entities in the same hierarchical format entity sets may have the same property values. To reduce the size of the property map representation, this document introduces an approach called "Property Inheritance". Individual entities can inherit the property from its hierarchical format entity set.

3. Protocol Specification: Basic Data Type

3.1. Entity Domain

3.1.1. Entity Domain Type

An entity domain has a type, which is defined 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). For example, the strings "ipv4", "ipv6", and "pid" are valid entity domain types.

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

An entity domain type defines the semantics of a type of entity domains. Each entity domain type MUST be registered with the IANA. The format of the entity identifiers (see Section 3.1.3) in that type of entity domains, as well as any hierarchical or inheritance rules (see Section 3.1.4) for those entities, MUST be specified at the same time.

3.1.2. Entity Domain Name

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

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

This document distinguish three types of entity domains: resource-specific entity domains, self-defined entity domain and aggregated entity domains. Their entity domain names are derived as follows.

Each ALTO information resource MAY define a resource-specific entity domain (which could be empty) in a given entity domain type. A resource-specific entity domain is identified by an entity domain name derived as follows. It MUST start with a resource ID using the ResourceID type defined in [RFC7285], followed by the "." separator (U+002E), followed by an EntityDomainType typed string. For example, if an ALTO server provides two network maps "netmap-1" and "netmap-2", they can define two different "pid" domains identified by "netmap-1.pid" and "netmap-2.pid" respectively. To be simplified, in the scope of a specific information resource, the resource-specific entity domain defined by itself can be identified by the "." EntityDomainType without the ResourceID.

When the associated information resource of a resource-specific entity domain is the current information resource itself, this resource-specific entity domain is a self-defined entity domain, and its ResourceID SHOULD be ignored from its entity domain name.

Given a set of ALTO information resources, there MAY be an aggregated entity domain in a given entity domain type amongst them. An aggregated entity domain is simply identified by its entity domain type. For example, given two network maps "net-map-1" and "net-map-2", "ipv4" and "ipv6" identify two aggregated Internet address entity domains (see Section 4.1) between them.

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

3.1.3. 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 registering a new entity domain type. Identifiers MAY be hierarchical, and properties MAY be inherited based on that hierarchy. Again, 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 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:db8:0:0:0:0:1" refer to the same entity in the "ipv6" entity domain.

3.1.4. Hierarchy and Inheritance

To make the representation efficient, some types of entity domains MAY allow the ALTO client/server to use a hierarchical format entity identifier to represent a block of individual entities. e.g., In an IPv4 domain "net1.ipv4", a cidr "net1.ipv4:192.0.2.0/26" represents 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.

3.2. 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 be identified. One entity MAY have multiple properties in the same type.

3.2.1. 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).

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.

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 a entity property may also depend on the 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.

3.2.2. 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.

3.3. Information Resource Export

Each information resource MAY export a set of entity domains and entity property mappings.

3.3.1. Resource-Specific Entity Domain Export

Each type of information resource MAY export several types of entity domains. For example, a network map resource defines a "pid" domain, a "ipv4" domain and a "ipv6" domain (which may be empty).

When a new ALTO information resource type is registered, if this type of information resource can export an existing type of entity domain, the corresponding document MUST define how to export such type of entity domain from such type of information resource.

When a new entity domain type is defined, if an existing type of information resource can export an entity domain in this entity domain type, the corresponding document MUST define how to export such type of entity domain from such type of information resource.

3.3.2. Entity Property Mapping Export

For each entity domain which could be exported by an information resource, this information resource MAY also export some mapping from this entity domain to some entity property. For example, a network map resource can map an "ipv4" entity to its "pid" property.

When a new ALTO information resource type is registered, if this type of information resource can export an entity domain in an existing entity domain type, and map entities in this entity domain to an existing type of entity property, the corresponding document MUST define how to export such type of an entity property.

When a new ALTO entity domain type or a new entity property type is defined, if an existing type of resource can export an entity domain in this entity domain type, and map entities in this entity domain to this type of entity property, the corresponding document MUST define how to export such type of an entity property.

4. Entity Domain Types

This document defines three entity domain types. The definition of each entity domain type below includes the following: (1) entity domain type name, (2) entity domain-specific entity identifiers, and (3) hierarchy and inheritance semantics. Since a global entity domain type defines a single global entity domain, we say entity domain instead of entity domain type.

4.1. Internet Address Domain Types

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

4.1.1. IPv4 Domain

4.1.1.1. Entity Domain Type

ipv4

4.1.1.2. Domain-Specific Entity Identifiers

Individual addresses are strings as specified by the IPv4Addresses rule of Section 3.2.2 of [RFC3986]; blocks of addresses are prefix-match strings as specified in Section 3.1 of [RFC4632]. For the purpose of defining 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.

4.1.2. IPv6 Domain

4.1.2.1. Entity Domain Type

ipv6

4.1.2.2. Domain-Specific Entity Identifiers

Individual addresses are strings as specified by Section 4 of [RFC5952]; blocks of addresses are prefix-match strings as specified in Section 7 of [RFC5952]. For the purpose of defining 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.

4.1.3. 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 some block C which prefix-matches I, then the address I inherits the value of P defined for block C. If more than one such block defines a value for P, I inherits the value of P in the block with the longest prefix. It is important to notice that this longest prefix rule will ensure no multiple inheritance, and hence no ambiguity.

Address blocks can also inherit properties: if a property P is not defined for a block C, but is defined for some block C' which covers all IP addresses in C, and C' has a shorter mask than C, then block C inherits the property from C'. If there are several such blocks C', C inherits from the block with the longest prefix.

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."
- o 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.

4.2. 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.

4.2.1. Entity Domain Type

pid

4.2.2. Domain-Specific Entity Identifiers

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

4.2.3. Hierarchy and Inheritance

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

4.2.4. 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".

4.3. 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, blocks of 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 a provider's network, would best be associated with the PID domain.

5. Entity Domains and Property Mappings in Information Resources

5.1. Network Map Resource

The ALTO network map resource defined by the media type "application/alto-networkmap+json" exports the following types of entity domains and entity property mappings.

5.1.1. Resource-Specific Entity Domain

An ALTO network map resource defines a "pid" domain, an "ipv4" domain and an "ipv6" domain by follows:

- o The defined "pid" domain includes all PIDs in keys of the "network-map" object.
- o The defined "ipv4" domain includes all IPv4 addresses appearing in the "ipv4" field of the endpoint address group of each PID.
- o The defined "ipv6" domain includes all IPv6 addresses appearing in the "ipv6" field of the endpoint address group of each PID.

5.1.2. Entity Property Mapping

For each of the preceding entity domains, an ALTO network map resource provides the properties mapping as follows:

ipv4 -> pid: An "networkmap" typed resource can map an "ipv4" entity to a "pid" property whose value is a PID defined by this "networkmap" resource and including the IPv4 address of this entity.

ipv6 -> pid: An "networkmap" typed resource can map an "ipv6" entity to a "pid" property whose value is a PID defined by this "networkmap" resource and including the IPv6 address of this entity.

5.2. Endpoint Property Resource

The ALTO endpoint property resource defined by the media type "application/alto-endpointprop+json" exports the following types of entity domains and entity property mappings.

5.2.1. Resource-Specific Entity Domain

An ALTO endpoint property resource defined an "ipv4" domain and an "ipv6" domain by follows:

- o The defined "ipv4" domain includes all IPv4 addresses appearing in keys of the "endpoint-properties" object.
- o The defined "ipv6" domain includes all IPv6 addresses appearing in keys of the "endpoint-properties" object.

5.2.2. Entity Property Mapping

For each of the preceding entity domains, an ALTO endpoint property resource exports the properties mapping from it to each supported global endpoint property. The property value is the corresponding global endpoint property value in the "endpoint-properties" object.

5.3. Property Map Resource

To avoid the nested reference and its potential complexity, this document does not specify the export rule of resource-specific entity domain and entity property mapping for the ALTO property map resource defined by the media type "application/alto-propmap+json" (see Section 6.1).

6. 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 9.4 gives an example of a property map request and its response.

6.1. Media Type

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

6.2. HTTP Method

The property map is requested using the HTTP GET method.

6.3. Accept Input Parameters

None.

6.4. 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.

6.5. 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).

6.6. 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.

7. 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 9.5, Section 9.6, Section 9.7 and Section 9.8 give examples of filtered property map requests and responses.

7.1. Media Type

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

7.2. HTTP Method

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

7.3. 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 7.4).

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 7.4). 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.

7.4. Capabilities

The capabilities are defined by an object of type PropertyMapCapabilities, as defined in Section 6.4.

7.5. Uses

Same to the "uses" field of the Property Map resource (see Section 6.5).

7.6. 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:

- o 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.

- o 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 6.6), except that:

- o 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.
- o 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.
- o 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., an "ipv4" or "ipv6" address block), 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:

- o 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.
- o 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.
- o 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. Impact on Legacy ALTO Servers and ALTO Clients

8.1. 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.

8.2. 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 do not distinguish between those two types of properties. Instead, if there

is a dependency, it is indicated by the "uses" capability of a property map, and is shared by all properties and entity domains in that map. Accordingly, it is RECOMMENDED that resource-specific endpoint properties be deprecated, and no new resource-specific endpoint properties be defined.

8.3. Impact on the pid Property

Section 7.1.1 of [RFC7285] defines the resource-specific endpoint property name "pid", whose value is the name of the PID containing that endpoint. For compatibility with legacy clients, an ALTO server which provides the "pid" property via the EPS MUST use that definition, and that syntax.

However, when used with property maps, this document amends the definition of the "pid" property as follows.

First, the name of the property is simply "pid"; the name is not prefixed with the resource ID of a network map. The "uses" capability of the property map indicates the associated network map. This implies that a property map can only return the "pid" property for one network map; if an ALTO server provides several network maps, it MUST provide a Property Map for each of the network maps.

Second, a client MAY request the "pid" property for a block of Internet addresses. An ALTO server determines the value of "pid" for an address block C as the rules defined in Section 7.6.

Note that although an ALTO server MAY provide a GET-mode property map which returns the entire map for the "pid" property, there is no need to do so, because that map is simply the inverse of the network map.

8.4. 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 blocks of identifiers, rather than just individual identifiers, which might change the semantics of a property.

9. Examples

9.1. 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:         ipv4:192.0.2.0/28  ipv4:192.0.2.16/28
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:         ipv4:192.0.2.0/28  ipv4:192.0.2.16/28
pid2:         ipv4:192.0.3.0/28  ipv4:192.0.3.16/28

```

Figure 4: Example Alternative Network Map

9.2. 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 |
| ipv4:192.0.2.0: | - | - | - | 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 "ipv4:192.0.2.0" 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

9.3. 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.

```
"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/full/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.exmaple.com/propmap/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" ]
    }
  }
}

```

Figure 8: Example IRD

9.4. Property Map Example

The following example uses the properties and IRD defined above 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"}
  }
}
```

9.5. 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"}
  }
}
```

9.6. 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"}
  }
}
```

9.7. Filtered Property Map Example #3

The following example uses the filtered property map resource to request the "pid" property for several IPv4 addresses and prefixes.

Note that the entity "ipv4:192.0.3.0/27" is redundant in the response. Although it can inherit a value of "defaultpid" for the

"pid" property from the entity "ipv4:0.0.0.0/0", none of addresses in it is in "defaultpid". Because blocks "ipv4:192.0.3.0/28" and "ipv4:192.0.3.16/28" have already cover all addresses in that block. So an ALTO server who wants a compact response can omit this entity.

```
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",
    "ipv4:192.0.3.0/27" ],
  "properties" : [ "default-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"},
    "ipv4:192.0.2.0/27": {"default-network-map.pid": "defaultpid"},
    "ipv4:192.0.3.0/28": {"default-network-map.pid": "pid3"},
    "ipv4:192.0.3.16/28": {"default-network-map.pid": "pid4"}
  }
}
```

9.8. 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. 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.

11. IANA Considerations

This document defines additional application/alto-* media types, and extends the ALTO endpoint property registry.

11.1. application/alto-* Media Types

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

| Type | Subtype | Specification |
|-------------|-------------------------|---------------|
| application | alto-propmap+json | Section 6.1 |
| application | alto-propmapparams+json | Section 7.3 |

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).

11.2. 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 4.1.1 | See Section 4.1.3 |
| ipv6 | See Section 4.1.2 | See Section 4.1.3 |
| pid | See Section 4.2 | 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.

11.2.1. Consistency Procedure between ALTO Address Type Registry and ALTO Entity Domain Registry

One potential issue of introducing the "ALTO Entity Domain Registry" is its relationship with the "ALTO Address Types Registry" already defined in Section 14.4 of [RFC7285]. In particular, the entity identifier of an entity domain registered in the "ALTO Entity Domain 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 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 is registered in the ALTO Entity Domain Registry.
- o If an ALTO entity domain 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 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 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 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 identifier to be the found ALTO address type identifier.
 - If no: Define a new ALTO entity domain 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 identifier to register a new ALTO entity domain in the ALTO Entity Domain Registry following Section 11.2.2 of this document.
- * If no (e.g., pid name, ane name or country code): Proceed with the ALTO Entity Domain registration as described in Section 11.2.2.

11.2.2. ALTO Entity Domain Registration Process

New ALTO entity domains are assigned after IETF Review [RFC5226] to ensure that proper documentation regarding the new ALTO entity domains and their security considerations has been provided. RFCs defining new entity domains SHOULD indicate how an entity in a registered 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 identifiers MUST conform to the syntactical requirements specified in Section 3.1.2. 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.
- o Entity Identifier Encoding: The procedure for encoding the identifier of an entity of the registered type as an EntityId (see Section 3.1.3). 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 Mapping to ALTO Address Type: A boolean value to indicate if the entity domain can be mapped to the ALTO address type with the same identifier.
- o 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.

11.3. ALTO Entity Property Type Registry

This document requests IANA to create and maintain the "ALTO Entity Property Type Registry", listed in Table 3.

To distinguish with the "ALTO Endpoint Property Type Registry", each entry in this registry is an ALTO entity property type defined in Section 3.2.1. Thus, registered ALTO entity property type identifier MUST conform to the syntactical requirements specified in that section.

The initial registered ALTO entity property types are listed in Table 3.

| 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 3.2.1 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.

This document requests registration of the identifier "pid", as shown in Table 3.

11.4. ALTO Resource-Specific Entity Domain Registries

11.4.1. Network Map

Media-type: application/alto-networkmap+json

| Entity Domain Type | Intended Semantics |
|--------------------|--------------------|
| ipv4 | See Section 5.1.1 |
| ipv6 | See Section 5.1.1 |
| pid | See Section 5.1.1 |

Table 4: ALTO Network Map Resource-Specific Entity Domain.

11.4.2. Endpoint Property

Media-type: application/alto-endpointprop+json

| Entity Domain Type | Intended Semantics |
|--------------------|--------------------|
| ipv4 | See Section 5.2.1 |
| ipv6 | See Section 5.2.1 |

Table 5: ALTO Endpoint Property Resource-Specific Entity Domain.

11.5. ALTO Resource Entity Property Mapping Registries

11.5.1. Network Map

Media-type: application/alto-networkmap+json

| Mapping Descriptor | Entity Domain Type | Property Type | Intended Semantics |
|--------------------|--------------------|---------------|--------------------|
| ipv4 -> pid | ipv4 | pid | See Section 5.1.2 |
| ipv6 -> pid | ipv6 | pid | See Section 5.1.2 |

Table 6: ALTO Network Map Entity Property Mapping.

12. 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.

13. 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>>.
- [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>>.

- [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>>.

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

ALTO WG
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

D. Lachos
C. Rothenberg
Unicamp
July 13, 2020

Multi-domain E2E Network Services
draft-lachosrothenberg-alto-md-e2e-ns-02

Abstract

Evolving networking scenarios (e.g., 5G) are considering the provision of value-added and on-demand end-to-end (E2E) network services in multi-domain (multi-operator/multi-technology) environments. This document presents different initiatives, mainly within standardization efforts and research projects, working on E2E network services across multiple domains. Problem statement and a layered network model are also described. In addition, this document raises an initial proposal towards a new ALTO service in support of E2E network service requirements. Finally, another important objective of this document is to begin a discussion about motivating use cases in scope of the ALTO WG after the re-chartering process.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 14, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 2 |
| 2. Context and Motivation | 3 |
| 2.1. Standardization Activities | 3 |
| 2.1.1. IETF | 3 |
| 2.1.2. ETSI | 4 |
| 2.1.3. MEF | 5 |
| 2.2. Research projects | 5 |
| 3. Problem Statement | 6 |
| 3.1. Network Function Placement Decisions | 6 |
| 3.2. Network Inventory | 6 |
| 3.3. Publishing Information | 6 |
| 4. Network Function Virtualization Architectures and Infrastructures | 7 |
| 4.1. Layered Network Model | 8 |
| 5. ALTO Extension: E2E Network Service Requirements Representation | 10 |
| 6. IANA Considerations | 12 |
| 7. Security Considerations | 12 |
| 8. Acknowledgments | 12 |
| 9. References | 12 |
| 9.1. Normative References | 12 |
| 9.2. Informative References | 12 |
| Authors' Addresses | 14 |

1. Introduction

The fifth generation (5G) of cellular networks is not only considered an evolution but a revolution in the field of information and communication technologies [WHITE-PAPER-5G]. 5G will support the creation of new and novel End-to-End (E2E) services, applications and complex use case scenarios, such as massive Internet of Things, extreme real-time communications, broadband access everywhere, higher user mobility. All these scenarios and services are triggering a modification in the way telecommunications sector deploy new network services, shifting from a commonly manual and long process to a flexible and programmable process.

In this context, cloud computing , Software Defined Networking (SDN), and Network Function Virtualization (NFV) arise as technological

pillars to achieve the necessary function programmability, network programmability, and resource virtualization during the provision of E2E network services.

The delivery of an E2E network service, or simply E2E service, often requires VNFs and their specific order [RFC7665]. Network operators start offering to their customers the possibility of configuring network services with specific requirements in terms of resources (e.g., cpu, memory, hard-disk) and performance objectives (e.g., bandwidth, latency) [VNF-PLAC]. Such demands are usually composed by distributed resources which are expected to be available across multiple domains with different technology and/or administration.

This document offers an overview of standardization activities and research projects, including problem statement, behind building E2E services traversing different domains (technological and/or administrative). Moreover, from a layered network model, it is proposed a potential ALTO extension related to E2E Network Service requirements representation based on the ETSI NFV MANO data model.

The overall rationale of this document is to arouse discussions into the ALTO WG concerning potential new items to be considered for the re-charter.

2. Context and Motivation

Different standardization efforts (e.g., IETF, MEF, ETSI) and research projects activities (e.g., 5GEX [H2020.5GEX], 5G-Transformer [H2020-5G-TRANSFORMER], T-NOVA [T-NOVA]) have been focused on multi-domain network service chaining. Standardization is essential to provide recommendations to create interoperable architectures with standardized protocols, and solutions (being developed by different projects) are addressing a diverse range of requirements to provide network services provided using multiple domains.

This section briefly describes, on the one hand, main standardization efforts delivering collections of norms and recommendations, while on the other hand it also provides an overview of several projects formed to develop network services across multiple domains.

2.1. Standardization Activities

2.1.1. IETF

SFC that span domains owned by single or multiple administrative entities are being proposed. The Hierarchical Service Function Chaining (hSFC) [RFC8459], for example, defines an architecture to deploy SFC in large networks. This RFC proposes to decompose the

network into smaller domains (domains under the control of a single organization). Another proposed initiative is [DRAFT-HH-MDSFC] that describes SFC crossing different domains owned by various organizations (e.g., ISPs) or by a single organization with administration partitions. The proposed architecture uses a SFC eXchange Platform (SXP) to collect and exchange information (topology, service states, policies, etc.) between different organizations and it works both in centralized (Multiple SFC domains connected by a logical SXP) and distributed (SXP server as a broker) environments.

More recently, the IETF ALTO WG started to discuss the uses of ALTO as an information model for representing network resource and services in multi-domain scenarios:

- o [DRAFT-ALTO-BROKER-MDO] proposes an ALTO-based Broker-assisted architecture where a broker plane works as a coordinator between a set of top-level control planes, i.e., Domain Orchestrators (DOs) and Multi-Domain Orchestrators (MdOs). The ALTO services (with the proposed extensions) provides abstract maps with a simplified, yet enough information view about MdOs involved in the federation. This information includes the abstract network topology, resource availability (e.g., CPUs, Memory, and Storage) and capabilities (e.g., supported network functions).
- o [DRAFT-ALTO-UNICORN] presents Unicorn, a resource orchestration framework for multi-domain, geo-distributed data analytics. This work resorts in ALTO as the information model to support the accurate, yet privacy-preserving resource discovery across different domains. The key information to be provided by the use of ALTO including different types of resources, e.g., the computing, storage, and networking resources.
- o [DRAFT-MD-SFC-ALTO] describes different standardization activities and research projects addressing the challenges posed by Service Function Chaining (SFC) across multiple domains (specifically, multiple administrative domains). In addition, this document presents an initial approach to realize inter-domain service chaining leveraging the ALTO protocol. Finally, another important concern of this document is to initiate a discussion (ALTO, SFC as well as 9other WGs) regarding if, how, and under what conditions ALTO can be useful to improve the multi-domain SFC process.

2.1.2. ETSI

The ETSI NFV ISG is paving the way toward viable architectural options supporting the efficient placement of functions in different administrative domains. More specifically, the document

[ETSI-NFV-IFA028] reports different NFV MANO architectural approaches with use cases related to network services provided using multiple administrative domains. Besides, it gives a non-exhaustive list of key information to be exchanged between administrative domains (monitoring parameters, topology view, resource capabilities, etc.) and recommendations related to security to permit the correct and proper operation of the final service.

2.1.3. MEF

With its work on the Service Operations Specification MEF 55 [MEF-SOE-MEF55], MEF has defined a reference architecture and framework for describing functional management entities (and interfaces between them) needed to support Lifecycle Service Orchestration (LSO). This LSO architecture enables automated management and control of E2E connectivity services across multiple operator networks. The automated service management includes fulfillment, control, performance, assurance, usage, security, analytics, and policy capabilities that make it possible, for example, expanding the footprint of service providers to interact with potentially several operators to manage and control the access portions of E2E services.

2.2. Research projects

Several projects include an architectural model integrating NFV management with SDN control capabilities to address the challenges towards flexible, dynamic, cost-effective, and on-demand service chaining.

[H2020.5GEX] aims to integrate multiple administrations and technologies through the collaboration between operators in the context of emerging 5G networking. [VITAL][T-NOVA] follow a centralized approach where each domain advertises its capabilities to a federation layer which will act as a broker. In order to avoid one network operator per country or regions, [H2020-5G-NORMA] proposes the use of management and control into a single virtual domain. Also, the 5G-Transformer project [H2020-5G-TRANSFORMER] is defining flexible slicing and federation of transport networking and computing resources across multiple domains. The NECOS project [H2020-NECOS], focused on the realization of E2E multi-domain cloud network slicing, proposes an architectural approach with slice information interfaces for resource exposure and resource discovery during slice provisioning. In addition, the architecture includes a slice marketplace interface between domain orchestrators and a marketplace broker.

3. Problem Statement

3.1. Network Function Placement Decisions

An E2E service request specify virtual nodes (a set of required VNFs) as well as virtual links (the order in which they must be executed). Virtual nodes are deployed in virtual machines hosted by different physical servers, and virtual links correspond to physical paths that connect those servers hosting VNFs. Both virtual nodes and virtual links are limited resources and both may also be located on different technological domains in a single administration and even crossing multiple administrations [VNF-MOB][SFC-ORC]. So that the placement decision problem involves to discover "best" candidate resources and "best" feasible paths between such resources.

3.2. Network Inventory

Placement decisions are a fundamental step for the management and orchestration of network services. Management systems (e.g., DOs, MdOs) need to maintain an inventory of the network providing a real-time representation or view of available infrastructure resources, software resources, and their relationships. However, The size of a network inventory can be very large in scenarios, such as distributed cloud and edge computing. As a result, management systems experiment scalability problems processing large amounts of data to decide where to instantiate a service or part of the service. Therefore, building a network inventory, under these circumstances, needs aggregation mechanisms to reduce time for discovery of resources and to simplify and optimize management of them.

3.3. Publishing Information

Once a network inventory is built, a mechanism for publishing information is also necessary so that the network inventory can provide a simplified, yet enough network information view to management systems. In order to retrieve such information to perform placement decisions, a communication protocol between management systems and network inventory is also necessary.

Therefore, on the one hand, network information (e.g., network locations, costs between them, endhost properties) needs to be advertised to the network applications and, on the other hand, network applications (e.g., DOs, MdOs) needs to describe their requirements and obtain information about resources that suit such requirements.

4. Network Function Virtualization Architectures and Infrastructures

With the introduction of NFV, network functions (e.g., switches, routers, firewalls), and also complex network functions (e.g., EPC) are able to be virtualized and implemented as a collection of virtual machines (VMs) deployed over the virtualized infrastructure. In turn, the virtualized infrastructure is instantiated on a substrate network.

In this context, one of the most accepted NFV architectural frameworks is the proposed by the ETSI ISG NFV working group [ETSI-NFV-WHITEPAPER]. Figure 1 [DRAFT-MD-VIRT] shows this NFV reference architecture. On the left, we can see the data plane: NFVIs hardware/software, VNFs, and optional element management systems. On the right, we see the control plane: VIM which is something like Openstack or Kubernetes, virtual network function managers, and the NFV Orchestrator on the top.

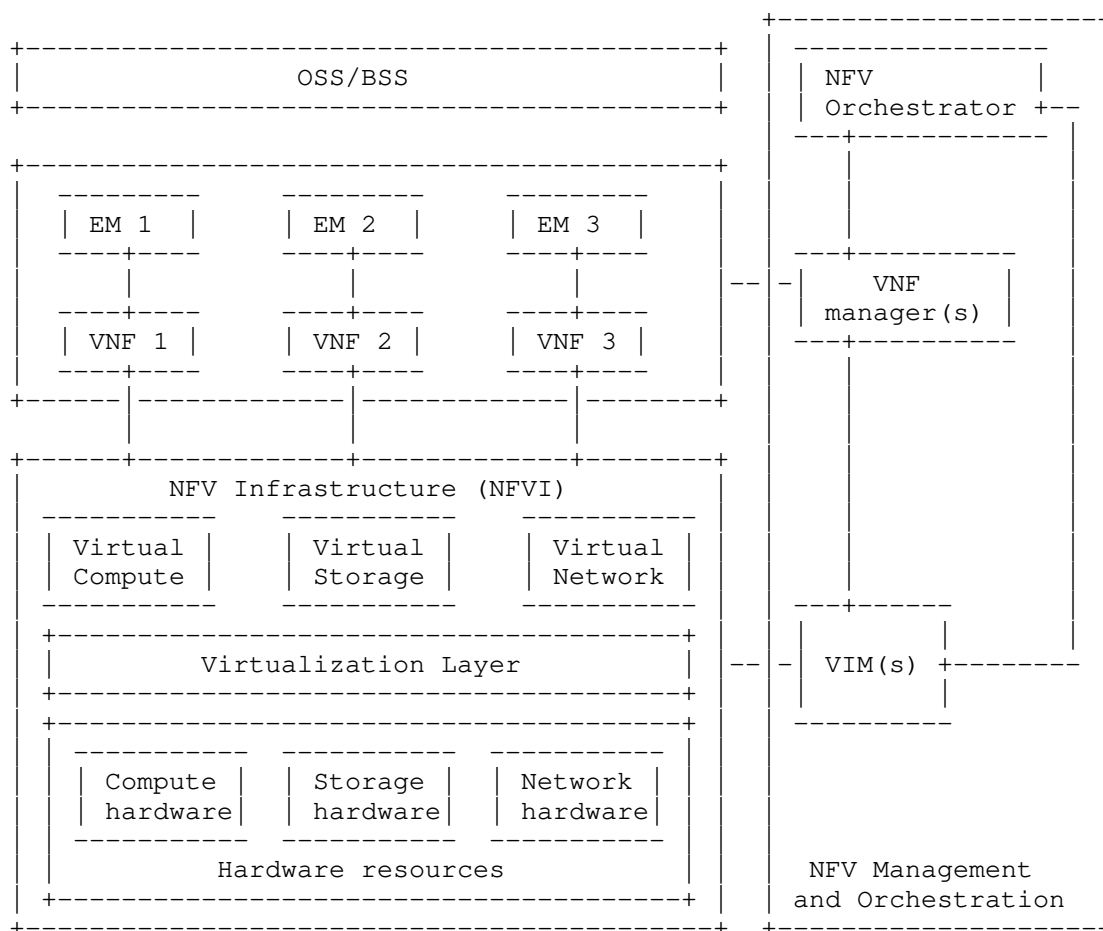


Figure 1: ETSI MANO Reference Architecture

4.1. Layered Network Model

Based on the ETSI NFV reference architecture, a layered network model is identified: Network Service layer, VNF layer, and Resource Layer. This model allows a separation of network relationships in different levels of abstraction. For example, network services can be queried at different levels of abstraction or we can map service paths in different layers (from an abstract to a more concrete layer).

In Figure 2, we have a network service with a set of interconnected VNFs. This network service topology is represented by the Network Service layer.

A VNF is typically divided into a set of virtual function components (VFCs) which comprise the VNF Layer. Each VFC is an application running within a single VM or container.

In case of the resource layer, we have virtual layer and physical layer. The virtual layer represents the virtual overlay network and the physical layer represents the substrate network. Virtualized infrastructures (e.g., VMs, virtual routers) are instantiated on a physical infrastructure.

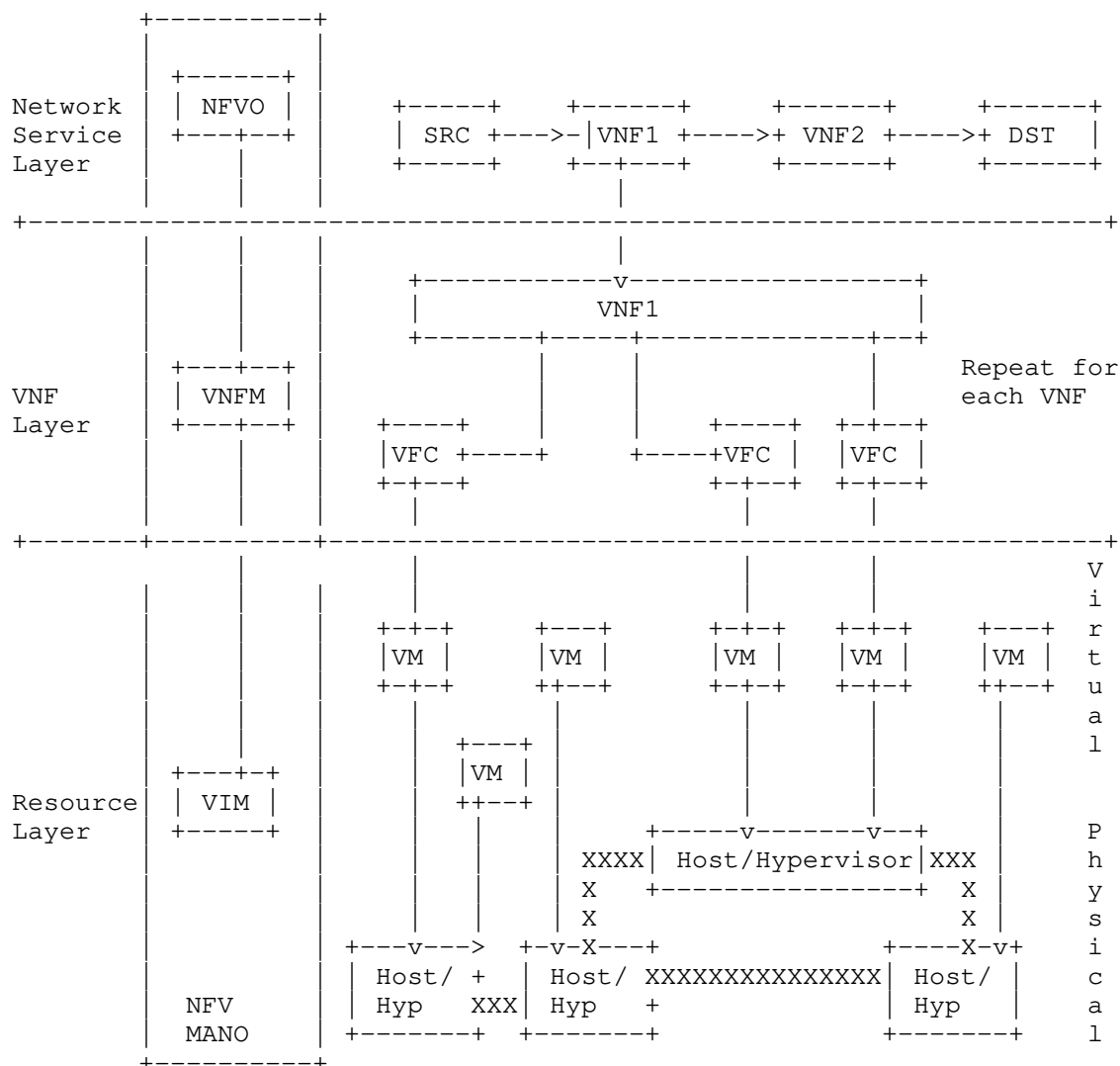


Figure 2: ETSI MANO Reference Architecture

5. ALTO Extension: E2E Network Service Requirements Representation

From the layered network model described in the previous section, we are considering an ALTO extension related to E2E Network Service requirements representation. An initial proposal has been presented in the ALTO-based Broker-assisted Mdo draft [DRAFT-ALTO-BROKER-MDO] where network applications (as ALTO clients) can specify a set of

basic E2E service requirements to an ALTO server in order to obtain candidate resources (domains) and candidate paths.

This initial E2E service requirement representation is inspired on the ETSI NFV MANO data model [ETSI-NFV-MAN001]. This model defines network services as a composition of network functions including the specification of deployment and operational requirements. Such specifications are captured in templates called Network Service Descriptor (NSD) and Virtual Network Function Descriptor (VNFD) that contain (relatively) static information used in the process of on-boarding network services and VNFs, respectively.

- o High level objects in a NSD include (among others) [ETSI-NFV-MAN001][OSM-DM]:
 - * Constituent VNFs: List of VNFDs that are part of the network service.
 - * VNF Dependencies: This describes dependencies between VNFs. For example, the order in which the VNFs inside a network service should be started.
 - * Network service Connection Points: Each network service has one or more external connection points (which act as endpoints) used to link two network services or to link external networks.
 - * Virtual Links: List of Virtual Link Descriptors (VLDs) that describe how VNFs (in the NSD) are connected.
- o High level objects in a VNFD include (among others) [ETSI-NFV-MAN001][OSM-DM]:
 - * Constituent VDUs: List of virtual deployment units (VDUs) in a specific VNF. Each VDU (also referred to VFC) describes the VM/Container capabilities (e.g., CPU, RAM, disks).
 - * VDU Dependencies: List of VDU dependencies used for determining the order of startup for VDUs.
 - * VNF Connection Points: List of external connection points used for connecting a VNF to other VNFs or to external networks.
 - * Internal VLDs: List of internal virtual links to connect various VDUs/VFCs.

6. IANA Considerations

This document includes no request to IANA.

7. Security Considerations

TBD.

8. Acknowledgments

This work is supported by the Innovation Center of Ericsson S.A., Brazil (grant agreement UNI.64). This document however does not necessary represent Ericsson's official viewpoint.

9. References

9.1. Normative References

- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8459] Dolson, D., Homma, S., Lopez, D., and M. Boucadair, "Hierarchical Service Function Chaining (hSFC)", RFC 8459, DOI 10.17487/RFC8459, September 2018, <<https://www.rfc-editor.org/info/rfc8459>>.

9.2. Informative References

- [DRAFT-ALTO-BROKER-MDO] Perez, D. and C. Rothenberg, "ALTO-based Broker-assisted Multi-domain Orchestration", draft-lachosrothenberg-alto-brokermdo-03 (work in progress), March 2020.
- [DRAFT-ALTO-UNICORN] Xiang, Q., Zhang, J., Le, F., Yang, Y., and H. Newman, "Resource Orchestration for Multi-Domain, Exascale, Geo-Distributed Data Analytics", draft-xiang-alto-multidomain-analytics-03 (work in progress), March 2020.
- [DRAFT-HH-MDSFC] Li, G., Li, G., Xu, Q., Zhou, H., and B. Feng, "Hybrid Hierarchical Multi-Domain Service Function chaining", draft-li-sfc-hhsfc-08 (work in progress), March 2020.

[DRAFT-MD-SFC-ALTO]

Perez, D., Xiang, Q., Rothenberg, C., and Y. Yang, "Multi-domain Service Function Chaining with ALTO", draft-lachos-multi-domain-sfc-alto-01 (work in progress), March 2020.

[DRAFT-MD-VIRT]

Bernardos, C., Contreras, L., Vaishnavi, I., Szabo, R., Li, X., Paolucci, F., Sgambelluri, A., Martini, B., Valcarenghi, L., Landi, G., Andrushko, D., and A. Mourad, "Multi-domain Network Virtualization", draft-bernardos-nfvrg-multidomain-05 (work in progress), September 2018.

[ETSI-NFV-IFA028]

ETSI, "Report on architecture options to support multiple administrative domains V3.1.1", Jan 2018, <http://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/028/03.01.01_60/gr_NFV-IFA028v030101p.pdf>.

[ETSI-NFV-MAN001]

ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration", Dec 2014, <https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf>.

[ETSI-NFV-WHITEPAPER]

ETSI, "Network Functions Virtualisation - White Paper 2", Oct 2013, <https://portal.etsi.org/NFV/NFV_White_Paper2.pdf>.

[H2020-5G-NORMA]

H2020, "5G-NORMA -- 5G Novel Radio Multiservice adaptive network Architecture", 2015, <<https://5gnorma.5g-ppp.eu/>>.

[H2020-5G-TRANSFORMER]

H2020, "5G-Transformer -- 5G Mobile Transport Platform for Vertical", 2017, <<http://5g-transformer.eu/>>.

[H2020-NECOS]

H2020 EU-Brazil, "NECOS -- Novel Enablers for Cloud Slicing", 2018, <<http://www.h2020-necos.eu/>>.

[H2020.5GEX]

Bernardos, C., Dugeon, O., Galis, A., Morris, D., Simon, C., and R. Szabo, "5G Exchange (5GEx)-- Multi-domain Orchestration for Software Defined Infrastructures", focus vol. 4, no.5, p.2, 2015.

- [MEF-SOE-MEF55] Metro Ethernet Forum, "Lifecycle Service Orchestration (LSO): Reference Architecture and Framework", Mar 2016, <https://www.mef.net/Assets/Technical_Specifications/PDF/MEF_55.pdf>.
- [OSM-DM] Open Source MANO, "OSM - Data Model", 2016, <https://osm.etsi.org/wikipub/index.php/Release_0_Data_Model_Details>.
- [SFC-ORC] Sun, G., Li, Y., Liao, D., and V. Chang, "Service Function Chain Orchestration across Multiple domains: A Full Mesh Aggregation Approach", IEEE Transactions on Network and Service Management 1175--1191, 2018.
- [T-NOVA] FP7 project T-NOVA, "T-NOVA Project, Network Functions as a Service over Virtualised Infrastructures", 2014, <<http://www.t-nova.eu/>>.
- [VITAL] VITAL PROJECT H2020, "VITAL -- Virtualized hybrid satellite-Terrestrial systems for resilient and flexible future networks", 2015, <<http://www.ict-vital.eu/>>.
- [VNF-MOB] Patel, A., Vutukuru, M., and D. Krishnaswamy, "Mobility-aware VNF placement in the LTE EPC", IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) 1--7, 2017.
- [VNF-PLAC] Slim, F., Guillemin, F., Gravey, A., and Y. Hadjadj-Aoul, "Towards a dynamic adaptive placement of virtual network functions under ONAP", IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) 210--215, 2017.
- [WHITE-PAPER-5G] NetWorld2020, ETP, "5g: Challenges, research priorities, and recommendations", Journal: Joint White Paper September, 2014.

Authors' Addresses

Danny Alex Lachos Perez
University of Campinas
Av. Albert Einstein 400
Campinas, Sao Paulo 13083-970
Brazil

Email: dlachosp@dca.fee.unicamp.br
URI: <https://intrig.dca.fee.unicamp.br/danny-lachos/>

Christian Esteve Rothenberg
University of Campinas
Av. Albert Einstein 400
Campinas, Sao Paulo 13083-970
Brazil

Email: chesteve@dca.fee.unicamp.br
URI: <https://intrig.dca.fee.unicamp.br/christian/>

ALTO WG
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

Q. Xiang
Yale University
F. Le
IBM
Y. Yang
Yale University
March 11, 2019

ALTO for Multi-Domain Applications: A Review of Use Cases and Design
Requirements
draft-xiang-alto-multidomain-usecases-00.txt

Abstract

With the development of novel network technology, such as software defined networking and network function virtualization, many novel multi-domain applications, such as flexible interdomain routing, distributed, federated machine learning and multi-domain collaborative dataset transfer, have been deployed. These applications can benefit substantially from the ALTO protocol [RFC7285], through which the information of multiple networks can be provided to applications. This document first introduces several multi-domain applications and how they can benefit from ALTO. It then describes a generic framework for multi-domain applications to use ALTO to improve the performance, followed by a discussion on new requirements and challenges for ALTO to better support these applications.

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 September 12, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|---|----|
| 1. Introduction | 2 |
| 2. Requirements Language | 3 |
| 3. Review of Multi-Domain Applications | 3 |
| 3.1. Flexible Interdomain Routing | 3 |
| 3.1.1. How flexible interdomain routing can benefit from ALTO? | 3 |
| 3.1.2. Example | 4 |
| 3.2. Resource Orchestration for Collaborative Data Sciences | 4 |
| 3.2.1. How multi-domain resource orchestration can benefit from ALTO | 4 |
| 3.2.2. Example | 5 |
| 3.3. Federated Machine Learning | 6 |
| 3.3.1. How federated machine learning can benefit from ALTO | 6 |
| 3.3.2. Example | 6 |
| 4. A Generic Framework | 7 |
| 4.1. Workflow | 8 |
| 5. Requirements of ALTO in Multi-Domain Applications | 9 |
| 5.1. Design Requirements | 9 |
| 5.2. Existing Efforts in the ALTO Working Group | 10 |
| 6. Summary | 10 |
| 7. References | 10 |
| 7.1. Normative References | 10 |
| 7.2. Informative References | 11 |
| Authors' Addresses | 12 |

1. Introduction

The ALTO protocol [RFC7285] provides network information to applications so that applications can make network informed decisions to improve the performance. Not only traditional applications such peer-to-peer systems, many recent, novel multi-domain applications,

which orchestrate resources across multiple networks, can also benefit substantially from ALTO.

The goal of this document is to explore how ALTO can help improve the performance of novel multi-domain applications, what ALTO extension services are needed, and what are the corresponding requirements and challenges for designing such extensions. To this end, this document first give a case-by-case review of emerging multi-domain applications and how they can benefit from ALTO. It then describes a generic framework for multi-domain applications to use ALTO to improve the performance, followed by a discussion on the need of new ALTO services and the corresponding requirements and challenges for these extensions to better support these applications.

2. 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].

3. Review of Multi-Domain Applications

3.1. Flexible Interdomain Routing

Flexible interdomain routing can be a highly valuable service for network providers. Specifically, an autonomous system (AS) providing such a service (the provider) allows other ASes (clients) to specify routing actions at the provider based on flexible matching conditions (e.g., match on TCP/IP 5-tuple). In this way, a client AS using the flexible interdomain routing service can offload access and traffic control to provider ASes, leading to a simpler client network configuration while giving the provider ASes additional business opportunities.

3.1.1. How flexible interdomain routing can benefit from ALTO?

ALTO provides provider ASes a standardized approach to expose its routing capability to client ASes. Traditional interdomain routing protocols such as BGP are not good options because they only expose the currently used routes, limiting client ASes' choices to specify flexible routes. In contrast, ALTO and its extensions provide interfaces for provider ASes to expose not only currently used routes, but also available yet unused routes, to client ASes so that they can have the flexibility to specify different routes for different data traffic.

3.1.2. Example

Consider the example in Figure 1. AS A is compromised and being used to send DDoS traffic to AS E. Without flexible interdomain routing, AS E can setup a firewall locally, but normal traffic from B to E will still be congested at C-D-E due to the existence of malicious traffic from A to E. If AS C provides flexible interdomain routing service, AS E can specify such a firewall at AS C to block DDoS traffic from A, and at the same time avoid the congestion of normal traffic from B to E.

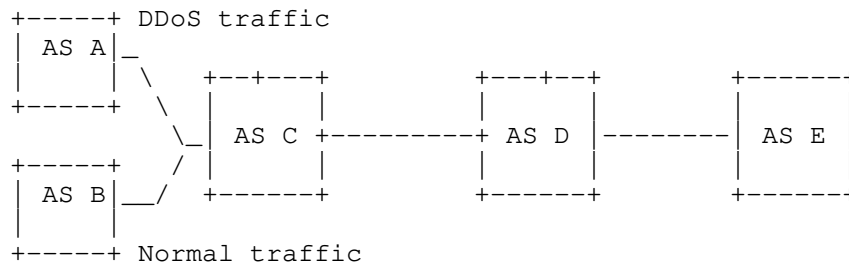


Figure 1: Flexible interdomain routing for DDoS mitigation.

3.2. Resource Orchestration for Collaborative Data Sciences

As the data volume increases exponentially over time, data analytics is transiting from a single-domain network to a multi-domain, geo-distributed network, where different member networks contribute various resources, e.g., computation, storage and networking resources, to collaboratively collect, share and analyze extremely large amounts of data. Such a paradigm calls for a unified resource orchestration framework to manage a large set of distributively-owned, heterogeneous resources, with the objective of efficient resource utilization, following the autonomy and privacy of different domains.

3.2.1. How multi-domain resource orchestration can benefit from ALTO

One key design challenge for multi-domain resource orchestration is its resource information model. Existing design options such as resource graph and ClassAds are inadequate because they cannot simultaneously (1) allow member networks to provide accurate information on different types of resource, (2) avoid the exposure of private information of member networks such as topology, and (3) allow data analytics jobs to accurately describe their requirements of different types of resources. In contrast, the section 7.1 of

Figure 2 discusses the advantages of choosing ALTO as the resource information model for multi-domain resource orchestration, and how ALTO can simultaneously satisfy the aforementioned design requirements.

3.2.2. Example

Consider an example of three member networks in Figure 2, where $s1$ and $s2$ are storage endpoints and $d1$ and $d2$ are computation endpoints. Assume a data analytics job is composed of two parallel tasks T1 and T2. T1 needs dataset X as input and T2 needs dataset Y as input.

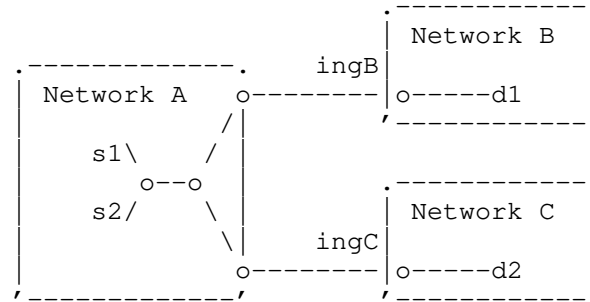


Figure 2: Multi-domain resource orchestration.

Using the ALTO endpoint property service, an ALTO client in the resource orchestrator can discover that $d1$ satisfies the computing requirements of T1 and $d2$ satisfies the computing requirements of T2. Hence there are only two candidate endpoint pairs: $(s1, d1)$ and $(s2, d2)$.

Afterwards, using the ALTO path vector extension, the ALTO client can retrieve the bandwidth sharing information of task T1 and T2, denoted as $x1$ and $x2$, respectively, as follows.

A: $x1 + x2 \leq 10\text{Mbps}$
 B: $x1 \leq 3\text{Mbps}$
 C: $x2 \leq 3\text{Mbps}$

With such information, the resource orchestrator can make the optimal resource orchestration decision to reserve 3 Mbps bandwidth for task T1, and 3 Mbps bandwidth for task T2.

3.3. Federated Machine Learning

Instead of moving large-scale datasets from multiple devices / networks to a centralized location for training, federated learning, is a distributed machine learning approach which enables training on distributed datasets residing on different autonomous systems (devices or networks). In this way, only updates on the training model need to be communicated between networks, leading to substantial reduction of networking resource consumption (e.g., saving bandwidth).

3.3.1. How federated machine learning can benefit from ALTO

Federated machine learning requires efficient scheduling algorithms to decide how networking resources should be allocated to transmit training model updates between different ASes. Similar as moving large-scale datasets between multiple ASes, moving updates of training model between ASes can also benefit from the availability of networking information, such as the AS-path and bandwidth sharing. ALTO provides a standardized approach for federated machine learning schedulers to retrieve such information from networks so that adaptive scheduling decisions can be made.

3.3.2. Example

Consider the example in Figure 3, where machine learning workers are located in AS A and D, while AS B and C are transit networks for data traffic transmitted between A and D. When AS A has a large, critical training model update to send to D. It first queries the ALTO servers at B and C for the endpoint cost (e.g., bandwidth) to transmit data from A to D. Suppose the ALTO server at AS B returns an endpoint cost of 10Mbps, while the ALTO server at AS C returns an endpoint cost of 100 Mbps. AS A can then use such information to make the optimal model update scheduling algorithm to send the training model update to AS D via AS C, instead of AS B.

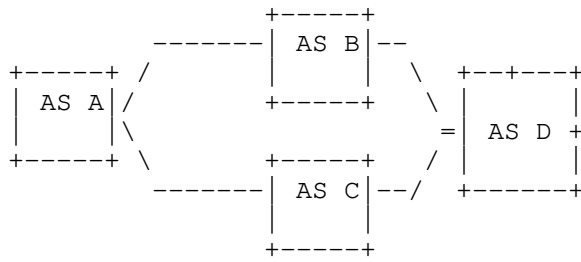


Figure 3: Federated machine learning.

4. A Generic Framework

After reviewing several important, novel multi-domain applications that can benefit substantially from ALTO, this document describes a generic framework for such applications to use ALTO to retrieve information from networks to improve their performance. The high-level architecture of this framework is given in Figure 4.

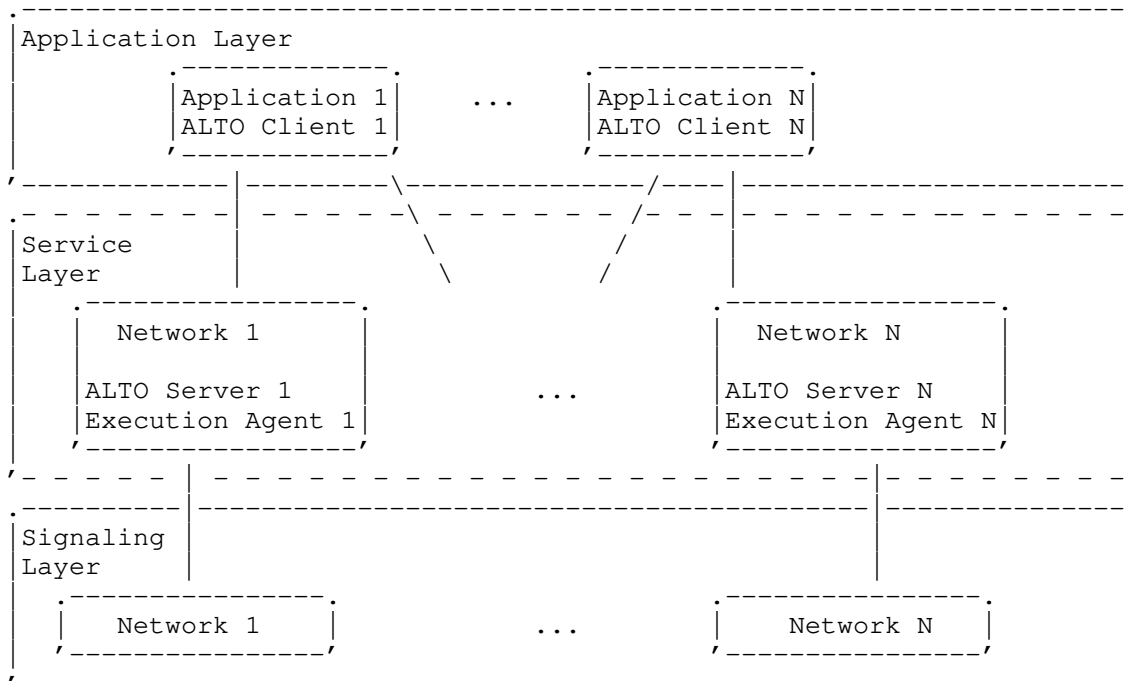


Figure 4: Generic framework of using ALTO in multi-domain applications.

The top layer of this framework is the application layer, in which each application deploy one or more ALTO clients to query for information provided by networks. The middle layer is the service layer. In this layer, each network deploys one more more ALTO servers to respond the queries sent by the ALTO clients from applications, and deploys one or more execution agents to respond to the applications' resource consumption actions. The bottom layer is the signaling layer, in which each network deploys interdomain protocols / systems, such as routing protocol BGP and resource reservation system OSCARS.

4.1. Workflow

The basic workflow of this framework is as follows.

- o An application identifies the networks whose resources (e.g., networking, computation and storage) it may want to consume, and invokes its ALTO clients to query the ALTO servers deployed in those networks for detailed resource information using base ALTO

protocol and its extension services (e.g., path vector, cost calendar and so on);

- o Upon receiving a query from an ALTO client, an ALTO server checks its local information, contacts the underlying signaling layer protocol / system of its residing network if local information is outdated, and returns the latest resource information to the querying ALTO client;
- o The applications uses the resource information collected from ALTO servers to make resource allocation decisions (e.g., route selection, resource reservation, etc.), and send such decisions to corresponding execute agents in the corresponding networks (e.g., the simple-reservation-interface of OSCARS).

5. Requirements of ALTO in Multi-Domain Applications

Using ALTO to improve the performance of recent novel multi-domain applications poses several new design requirements. This section discusses these requirements and briefly review existing efforts in the ALTO working group aiming to satisfy them.

5.1. Design Requirements

- o Exposing information of alternative resources. Current ALTO protocols and its extensions only provide information of currently used resources (e.g., currently used interdomain route). However, exposing information of alternative resources (e.g., available but not used interdomain routes) may provide the users of new multi-domain applications (e.g., flexible interdomain routing) more flexibility on choosing different resources, giving networks that provide such applications additional business opportunities.
- o Providing a unified, accurate representation of multiple types of resources. Current ALTO protocols and its extensions mainly focus on providing network information to applications, with the exception of endpoint property service. However, as new multi-domain applications often consume multiple types of resources across multiple networks, encoding such information accurately in a unified approach is crucial for deploying ALTO to improve such applications' performance.
- o Providing interfaces for more flexible query. Current ALTO protocol and its extensions allows applications to query resource information by specifying IP addresses of endpoints and simple filters. However, with the emerging of new networking architecture (e.g., software defined networking and network function virtualization) and the fine-grained resource requirement

of applications (e.g., link-disjoint paths and endpoint precedence), applications need a more flexible interface to specify queries of resource information.

5.2. Existing Efforts in the ALTO Working Group

Several documents have been submitted to the ALTO working group, with the aim to satisfy one or more of the design requirements discussed above. For example, [DRAFT-PV], [DRAFT-RSA], [DRAFT-UNICORN-INFO] and several other documents propose and apply the ALTO path vector extension to provide accurate networking resource information to support multi-domain resource orchestration. [DRAFT-NFCHAIN] proposes to use ALTO to support resource orchestration for multi-domain service function chaining, and proposes a new ALTO extension to retrieve AS path of network functions across different networks. [DRAFT-CONTEXT] proposes to extend cost information specified in RFC7285 by providing several possible cost values for the same cost metric where each value depends on qualitative criteria as opposed to quantitative criteria such as time. [DRAFT-UR] makes a proposal to use mathematical programming constraint as a generic representation of multiple resources. [DRAFT-FCS] proposes a flexible flow query extension service to allow applications to specify query entities based on flexible matching conditions (e.g., TCP/IP 5-tuple) instead of IP addresses only.

6. Summary

This document reviews several emerging multi-domain applications and how they can benefit from ALTO. It then describes a generic framework for multi-domain applications to use ALTO to improve the performance. In addition, several design requirements are discussed. Though different drafts in the working group have been trying to address one or more these design requirements, a systematic investigation of these issues is still missing. The authors of this document plan to perform such an investigation and make a unified design proposal in the next version of this document.

7. References

7.1. 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>>.

7.2. Informative References

[DRAFT-CONTEXT]

Randriamasy, S., "ALTO Contextual Cost Values", 2017, <<https://datatracker.ietf.org/doc/draft-randriamasy-alto-cost-context/>>.

[DRAFT-FCS]

Zhang, J., Gao, K., Wang, J., Xiang, Q., and Y. Yang, "ALTO Extension: Flow-based Cost Query", 2017, <<https://datatracker.ietf.org/doc/draft-gao-alto-fcs/>>.

[DRAFT-NFCHAIN]

Perez, D. and C. Rothenberg, "ALTO-based Broker-assisted Multi-domain Orchestration", 2018, <<https://datatracker.ietf.org/doc/html/draft-lachosrothenberg-alto-brokermdo-01>>.

[DRAFT-PV]

Bernstein, G., Lee, Y., Roome, W., Scharf, M., and Y. Yang, "ALTO Extension: Abstract Path Vector as a Cost Mode", 2015, <<https://tools.ietf.org/html/draft-yang-alto-path-vector-01>>.

[DRAFT-RSA]

Gao, K., Wang, X., Xiang, Q., Gu, C., Yang, Y., and G. Chen, "A Recommendation for Compressing ALTO Path Vectors", 2017, <<https://datatracker.ietf.org/doc/draft-gao-alto-routing-state-abstraction/>>.

[DRAFT-UNICORN-INFO]

Xiang, Q., Newman, H., Bernstein, G., Du, H., Gao, K., Mughal, A., Balcas, J., Zhang, J., and Y. Yang, "Implementation and Deployment of A Resource Orchestration System for Multi-Domain Data Analytics", 2017, <<https://datatracker.ietf.org/doc/draft-xiang-alto-exascale-network-optimization/>>.

[DRAFT-UP]

Roome, W., Chen, S., Randriamasy, S., Yang, Y., and J. Zhang, "Unified Properties for the ALTO Protocol", 2015, <<https://datatracker.ietf.org/doc/draft-ietf-alto-unified-props-new/>>.

[DRAFT-UR]

Xiang, Q., Le, F., and Y. Yang, "ALTO Extension: Unified Resource Representation", 2018,
<<https://datatracker.ietf.org/doc/draft-xiang-alto-exascale-network-optimization/>>.

[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>>.

[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>>.

Authors' Addresses

Qiao Xiang
Yale University
51 Prospect Street
New Haven, CT
USA

Email: qiao.xiang@cs.yale.edu

Franck Le
IBM
Thomas J. Watson Research Center
Yorktown Heights, NY
USA

Email: fle@us.ibm.com

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

Email: yry@cs.yale.edu